

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Комп'ютерних наук і технологій

(повне найменування факультету)

Системний аналіз та обчислювальна математика

(повне найменування кафедри)

Пояснювальна записка

до дипломної роботи

Бакалавр

(ступінь вищої освіти)

на тему Аналіз та прогнозування ефективності рекламних
(назва теми)
кампаній

Виконав(ла): студент(ка) 4 курсу, групи КНТ-813сп

Спеціальності 124 - Системний аналіз
(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інтелектуальні технології та прийняття рішень в
складних системах

ТАГАН О.С

(ПРИЗВИЩЕ та ініціали)

Керівник РЯБЕНКО А.Є

(ПРИЗВИЩЕ та ініціали)

Рецензент ПИРОЖОК А.В

(ПРИЗВИЩЕ та ініціали)

2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій

Кафедра Системного аналізу та обчислювальної математики

Ступінь вищої освіти бакалавр

Спеціальність 124 - Системний аналіз
(код і найменування)

Освітня програма (спеціалізація) Інтелектуальні технології та прийняття рішень
(назва освітньої програми (спеціалізації))

в складних системах

ЗАТВЕРДЖУЮ

Завідувач кафедри Системного аналізу та обчислювальної математики ТЕРЕЩЕНКО Е.В

« 08 » червня 20 26 року

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

ТАГАН Олесі Сергіївни

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Аналіз та прогнозування ефективності рекламних кампаній

керівник проєкту (роботи) к.ф.-м.н., доц., РЯБЕНКО Антон Євгенович

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 23 » квітня 20 26 №

2. Строк подання студентом проєкту (роботи) « 08 » червня 20 26 року

3. Вихідні дані до проєкту (роботи) Статистичні дані рекламних кампаній, що включають витрати на рекламу, кількість кліків, покупок, дохід та інші показники ефективності для кожного типу реклами. Також теоретичні відомості щодо оцінювання рекламних кампаній, benchmark-показники для порівняння результатів. Використовуються методи статистичного аналізу та прогнозування (z-score аналіз, кореляційний аналіз, прогнозування результатів кампаній за допомогою лінійної регресії та через рекламну воронку за песимістичним, базовим та оптимістичним сценаріями).

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно

розробити) аналіз предметної області та постановка задачі, дослідження методів оцінювання ефективності рекламних кампаній, опис методів статистичного аналізу та прогнозування, проектування та реалізація веб-застосунку, тестування програмного продукту, аналіз результатів роботи системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Презентація до дипломної роботи на 40 слайдів;скріншоти інтерфейсу веб-застосунку; візуалізація результатів аналізу рекламної кампанії; відеоролик демонстрації роботи веб-застосунку.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1	РЯБЕНКО А.Є., к.ф.-м.н., доц	22.04.2026	08.06.2026
2	РЯБЕНКО А.Є., к.ф.-м.н., доц	22.04.2026	08.06.2026
3	РЯБЕНКО А.Є., к.ф.-м.н., доц	22.04.2026	08.06.2026
НОРМКОНТРОЛЬ	ШИРОКОРАД Д.В., к.ф.-м.н., доц	08.06.2026	

7. Дата видачі завдання « 22 » квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Вибір теми дипломної роботи	28.11.2025 - 26.03.2026	
2	Визначення функціоналу веб-застосунку та вибір технологій для реалізації	30.03.2026 - 06.04.2026	
3	Розробка веб-застосунку	08.04.2026 - 28.05.2026	
4	Створення презентації до дипломної роботи	25.05.2026- 28.05.2026	
6	Попередній захист дипломної роботи	29.05.2026	
7	Захист дипломної роботи	10.06.2026	

Студент(ка)

_____ (підпис)

Олеся ТАГАН

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ (підпис)

АНТОН РЯБЕНКО

РЕФЕРАТ

Дипломна робота: 162 с., 43 рис., 1 дод.

Пояснювальна записка до диплому:

Мета дипломної роботи – розробити веб-застосунок для аналізу та прогнозування ефективності рекламних кампаній з автоматичним розрахунком показників, порівняння результатів і формування рекомендацій.

Об'єкт дослідження – рекламні кампанії та процес оцінювання їх ефективності.

Предмет дослідження – методи аналізу, порівняння та прогнозування рекламних показників.

У дипломній роботі було створено веб-застосунок, який дозволяє користувачу вводити дані рекламної кампанії вручну або завантажувати їх із CSV-файлу. Система автоматично розраховує ключові метрики ефективності рекламних кампаній, порівнює кампанії та періоди між собою. Також у веб-застосунку використовуються методи статистичного аналізу, таких як Z-score та кореляційний аналіз, а також способи прогнозування результатів кампаній через рекламну воронку і лінійну регресію. Веб-застосунок будує графіки, формує рекомендації та прогнозує можливі результати рекламної кампанії.

РЕКЛАМНА КАМПАНІЯ, РЕКЛАМНІ МЕТРИКИ ЕФЕКТИВНОСТІ, АНАЛІЗ ДАНИХ, Z-SCORE, КОРЕЛЯЦІЙНИЙ АНАЛІЗ, ЛІНІЙНА РЕГРЕСІЯ, ПРОГНОЗУВАННЯ, JAVASCRIPT

ЗМІСТ

ЗАВДАННЯ	2
РЕФЕРАТ	4
ВСТУП	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Огляд предметної області цифрової реклами	9
1.2.1 Базові метрики, спільні для всіх типів кампаній	11
1.2.2 Метрики для кампаній «Продажі»	13
1.2.3 Метрики для кампаній «Лідогенерація»	16
1.2.4 Метрики для кампаній «Трафік»	17
1.2.5 Метрики для кампаній «Охоплення»	17
1.2.6 Метрики для кампаній «Залучення»	18
1.3 Benchmark-показники як орієнтир для оцінювання рекламних кампаній	19
1.4 Аналіз існуючих рішень для рекламної аналітики	21
1.5 Постановка задачі та вимоги до веб-застосунку	23
РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ АНАЛІЗУ РЕКЛАМНИХ КАМПАНІЙ	25
2.1 Z-score для виявлення відхилень	25
2.2 Кореляція Пірсона для аналізу зв'язків між показниками	27
2.3 Методи прогнозування результатів рекламної кампанії	30
2.3.1 Прогноз через рекламну воронку	30
2.3.2 Прогноз через лінійну регресію	33
2.3.3 Сценарний прогноз	35
2.4 Методика комплексної оцінки Campaign Score	38
2.4.1 Загальна логіка розрахунку	38
2.4.2 Ваги метрик	39
2.4.3 Особливості оцінювання Sales-кампаній	40
2.4.4 Корекція змішаних сигналів	41

2.4.5	Корекція змішаних сигналів	42
2.5	Формування рекомендацій на основі результатів аналізу	42
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ		45
3.1	Вибір технологій та середовища розробки	45
3.2	Загальна структура веб-застосунку	47
3.3	Реалізація введення, обробки та перевірки даних	51
3.3.1	Вибір параметрів аналізу	51
3.3.2	Ручне введення даних	55
3.3.3	Завантаження та обробка CSV-файлів	59
3.3.4	Перевірка коректності даних	64
3.3.5	Нормалізація даних перед аналізом	68
3.4	Реалізація розрахунку рекламних метрик та аналітичних показників	71
3.4.1	Розрахунок рекламних метрик за типами кампаній	71
3.4.2	Захист від некоректних обчислень	72
3.4.3	Реалізація benchmark-порівняння та статусів показників	77
3.4.4	Реалізація статистичного аналізу	84
3.4.5	Реалізація Campaign Score	90
3.5	Реалізація модуля прогнозування	100
3.5.1	Вхідні параметри прогнозу	100
3.5.2	Прогноз через рекламну воронку	104
3.5.3	Прогноз на основі лінійної регресії	111
3.5.4	Сценарний прогноз	117
3.6	Реалізація візуалізації результатів за допомогою графіків	121
РОЗДІЛ 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ		128
4.1	Мета та підхід до тестування веб-застосунку	128
4.2	Тестування введення та перевірки даних	130
4.3	Перевірка правильності розрахунку рекламних метрик	133
4.4	Тестування статистичного аналізу та Campaign Score	137
4.5	Тестування модуля прогнозування	141

4.6 Тестування візуалізації результатів	145
4.7 Тестування блоку діагностичних рекомендацій	148
4.8 Аналіз результатів тестування та обмеження системи	151
ВИСНОВКИ	153
ПЕРЕЛІК ПОСИЛАНЬ	155
ДОДАТОК А Фрагменти програмного коду веб-застосунку	156

ВСТУП

У наш час реклама є одним із ключових інструментів залучення клієнтів. І через це ми постійно бачимо рекламу в інтернеті, що стало звичайною частиною повсякденного життя. Щоб зрозуміти, наскільки ефективно працює реклама, потрібно аналізувати різні показники такі як CTR, CPC, CPA, ROAS, загальний дохід та інші метрики в залежності від типу рекламної кампанії.

Найчастіше аналіз рекламних кампаній виконується вручну. Це може призвести до помилок в розрахунках ключових метрик. Також робота займає більше часу, особливо якщо потрібно перевіряти не одну, а декілька кампаній одночасно. Крім того, під час ручного аналізу складно швидко порівнювати різні кампанії між собою або аналізувати різні періоди однієї кампанії. У рекламній сфері важливо вчасно помічати проблеми, адже витрати бюджету мають приносити результат, а не бути неприбутковими.

Тому задачею дипломної роботи є розробка веб-застосунку, який автоматично аналізує ефективність рекламних кампаній і прогнозує результати рекламних кампаній. Веб-застосунок передбачає можливість введення даних кампанії вручну через зручну форму або завантаження CSV-файлу з рекламного кабінету. Система автоматично обчислює основні рекламні метрики, порівнює кампанії та різні періоди, будує графіки для відображення результатів, формує рекомендації та прогнозує результати за різними можливими сценаріями.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд предметної області цифрової реклами

Цифрова реклама є одним із головних інструментів просування товарів і послуг у сучасному бізнесі. Її використовують у соцмережах, пошукових системах, відеоплатформах, месенджерах та на інших онлайн-майданчиках. На відміну від традиційної реклами, цифрова реклама дозволяє рекламодавцям отримувати інформацію про поведінку користувача. Це, наприклад, скільки користувачів переглянули рекламу, скільки натиснули на неї, скільки відбулося конверсій, дохід від реклами та інше. Завдяки цим даним можна зрозуміти наскільки ефективно працює реклама і чи досягає вона своєї мети.

Рекламні кампанії відрізняються за своїми цілями. Тому одні й ті самі показники не завжди однаково важливі для всіх кампаній. Наприклад, для кампанії де мета продажі ключовим результатом є показник ROAS (рентабельність рекламних витрат) та скільки чистого прибутку це принесло, тоді як для кампанії, які працюють на охоплення, головною ціллю виступає скільки унікальних користувачів побачили рекламу, та частота показів.

У цій дипломній роботі розглядаються п'ять найпоширеніших типів рекламних кампаній: продажі (Sales), лідогенерація (Lead Generation), трафік (Traffic), охоплення (Awareness) та залучення (Engagement). Кожен із цих типів має свою ціль, тому для його оцінювання використовується свій набір ключових показників. Узагальнена характеристика типів рекламних кампаній наведена в таблиці 1.1.

Таблиця 1.1 — Типи рекламних кампаній та їх ключові метрики

Тип кампанії	Основна ціль кампанії	Ключові показники кампанії
Продажі	Кількість покупок та чистий дохід	ROAS, Profit, CPA, CR, AOV
Лідогенерація	Отримання заявок від потенційних клієнтів	CPL, Lead CR, Qualified Lead Rate
Трафік	Залучення користувачів на цільову сторінку	CTR, CPC, LPV Rate, Cost per LPV.
Охоплення	Збільшення кількості унікальних користувачів, які побачили рекламу	Reach, CPM, Cost per 1000 Reach, Frequency
Залучення	Отримання взаємодії із рекламним контентом	Engagement Rate, Cost per Engagement

Як видно з таблиці 1.1, оцінювання рекламної кампанії залежить від її цілі. Саме тому під час аналізу потрібно враховувати не один окремий показник, а загальну картину роботи кампанії. Детальне значення основних рекламних метрик та способи їх розрахунку розглянуто в наступному підрозділі.

1.2 Основні метрики рекламних кампаній та формули їх обчислення

Перед розробкою веб-застосунка для аналізу рекламних кампаній необхідно визначити, які саме метрики він буде обчислювати, від яких вхідних даних вони залежать і за якими формулами розраховуються. У цій роботі використовуються показники, які рекламодавець може ввести вручну або імпортувати з CSV-файлу, експортованого з рекламного кабінету.

До основних вхідних даних належать:

- Ad spend – загальні витрати на рекламу;
- Impressions – кількість показів;
- Clicks – кількість кліків;
- Reach – охоплення унікальних користувачів;
- Frequency – середня частота показів;
- Revenue – отриманий дохід;
- Purchases – кількість покупок;
- Leads – кількість заявок;
- Qualified Leads – кількість кваліфікованих заявок;
- Landing Page Views – перегляди цільової сторінки;
- Engagements – взаємодії з рекламним контентом;
- Comments, Shares, Saves – коментарі, поширення та збереження;
- Target Audience Size – розмір цільової аудиторії;
- Total Product Cost – повна собівартість проданих товарів;
- Average Gross Margin – середня валова маржа.

Усі похідні метрики обчислюються на основі цих вхідних показників [1].

Далі наведено основні метрики, які використовуються в розробленому застосунку.

1.2.1 Базові метрики, спільні для всіх типів кампаній

Базові метрики використовуються майже для всіх типів рекламних кампаній. Вони показують, як часто реклама демонструється користувачам, наскільки активно аудиторія реагує на оголошення і якою є загальна вартість рекламної взаємодії.

CTR (Click-Through Rate) – це відсоток користувачів, які після перегляду рекламного оголошення натиснули на нього. Цей показник допомагає оцінити, наскільки рекламне повідомлення привертає увагу аудиторії. Розраховується за формулою [1]:

$$CTR = \frac{Clicks}{Impressions} \cdot 100\% \quad (1.1)$$

CPC (Cost Per Click) – це середня ціна за клік, яку рекламодавець платить за перехід користувача за рекламним оголошенням. Розраховується за формулою [1]:

$$CPC = \frac{Ad\ spend}{Clicks} \quad (1.2)$$

CPM (Cost Per Mille) – це ціна, яку рекламодавець сплачує за тисячу показів рекламного оголошення. Розраховується за формулою [1]:

$$CPM = \frac{Ad\ spend}{Impressions} \cdot 1000 \quad (1.3)$$

Frequency (частота показів) – це середня кількість разів, яку один користувач побачив рекламне оголошення. Розраховується за формулою [1]:

$$Frequency = \frac{Impressions}{Reach} \quad (1.4)$$

Reach Rate (частка охопленої цільової аудиторії) – це відсоток цільової аудиторії, яку вдалося охопити рекламною кампанією. Розраховується за формулою [1]:

$$Reach Rate = \frac{Reach}{Target Audience Size} \cdot 100\% \quad (1.5)$$

1.2.2 Метрики для кампаній «Продажі»

Для кампаній типу «Продажі» основна увага приділяється тому, як рекламні витрати пов'язані з отриманим доходом і прибутком. Тому для таких кампаній використовуються метрики, які допомагають зрозуміти, чи окупилися рекламні витрати і чи є кампанія прибутковою.

ROAS (Return on Ad Spend, рентабельність рекламних витрат) — це відношення доходу, отриманого від кампанії, до витрат на рекламу. ROAS показує, скільки доходу принесла кожна одиниця рекламного бюджету. Розраховується за формулою [1]:

$$ROAS = \frac{Revenue}{Ad spend} \quad (1.6)$$

Gross Profit (валовий прибуток) – це різниця між доходом і собівартістю проданих товарів. У застосунку підтримується два способи розрахунку валового прибутку залежно від того, яким чином рекламодавець ввів дані про собівартість. Якщо введено повну суму собівартості, формула має вигляд:

$$Gross Profit = Revenue - TotalProductCost \quad (1.7)$$

Якщо ж рекламодавець знає лише середню валову маржу у відсотках, валовий прибуток обчислюється як [1]:

$$Gross Profit = Revenue \cdot \frac{AverageGrossMargin}{100} \quad (1.8)$$

Gross Margin (валова маржа) – показує, яка частина доходу залишається після покриття собівартості товарів, але ще до віднімання витрат на рекламу.

Формула:

$$Gross Margin = \frac{Gross Profit}{Revenue} \cdot 100\% \quad (1.9)$$

Profit (чистий прибуток рекламної кампанії) – це сума, яка залишається після віднімання собівартості товарів і рекламних витрат. Якщо відома повна собівартість проданих товарів, прибуток обчислюється за формулою:

$$Profit = Revenue - Total Product Cost - Ad Spend \quad (1.10)$$

Якщо валовий прибуток уже розраховано на основі маржі, прибуток можна визначити так:

$$Profit = Gross Profit - Ad Spend \quad (1.11)$$

Break-even ROAS (ROAS беззбитковості) – це мінімальний рівень ROAS, за якого реклама не приносить збитків. Якщо фактичний ROAS вищий за Break-even ROAS, кампанія є прибутковою. Якщо нижчий є збитковою. Формула:

$$BreakEven ROAS = \frac{1}{\frac{Gross Margin}{100}} \quad (1.12)$$

CPA (Cost Per Action) – це ціна, яку рекламодавець витрачає на отримання однієї покупки. Розраховується за формулою:

$$CPA = \frac{Ad\ Spend}{Purchases} \quad (1.13)$$

CR (Conversion Rate) – це відсоток кліків, які завершилися покупкою. Формула [1]:

$$CR = \frac{Purchases}{Clicks} \cdot 100\% \quad (1.14)$$

AOV (Average Order Value, середній чек) – це середня сума однієї покупки. Розраховується за формулою [1]:

$$AOV = \frac{Revenue}{Purchases} \quad (1.15)$$

Max Profitable CPA (максимально допустима вартість покупки) показує, скільки максимум можна витратити на залучення однієї покупки, щоб кампанія ще не стала збитковою. Формула:

$$Max\ Profitable\ CPA = \frac{Gross\ Profit}{Purchases} \quad (1.16)$$

1.2.3 Метрики для кампаній «Лідогенерація»

Для кампаній типу «Лідогенерація» ключовим є оцінити, скільки коштує одна заявка і яка її якість.

Lead CR (Lead Conversion Rate) – це частка кліків, які перетворилися на заявки. Чим вищий Lead CR, тим ефективніше працює форма заявки на сайті і тим релевантніший трафік приходить з реклами. Формула:

$$\text{Lead CR} = \frac{\text{Leads}}{\text{Clicks}} \cdot 100\% \quad (1.17)$$

CPL (Cost Per Lead, вартість однієї заявки) – це ціна, яку рекламодавець витрачає на отримання однієї заявки. Розраховується за формулою:

$$\text{CPL} = \frac{\text{Ad Spend}}{\text{Leads}} \quad (1.18)$$

Qualified Lead Rate (частка кваліфікованих заявок) – це відсоток заявок, які після перевірки можуть розглядатися як потенційно якісні клієнти. Розраховується за формулою:

$$\text{Qualified Lead Rate} = \frac{\text{Qualified Leads}}{\text{Leads}} \cdot 100\% \quad (1.19)$$

Cost per Qualified Lead (вартість однієї кваліфікованої заявки) – це ціна, витрачена на отримання однієї якісної заявки. Формула:

$$\text{Cost per Qualified Lead} = \frac{\text{Ad Spend}}{\text{Qualified Leads}} \quad (1.20)$$

1.2.4 Метрики для кампаній «Трафік»

Для кампаній типу Traffic важливо оцінити, наскільки ефективно реклама приводить користувачів на цільову сторінку.

LPV Rate (Landing Page View Rate, частка переглядів цільової сторінки) – це відсоток кліків, які завершилися реальним переглядом цільової сторінки. Різниця між кількістю кліків і кількістю переглядів сторінки може бути значною: користувач натискає на оголошення, але сторінка може довго завантажуватися, або користувач відразу закриває вкладку. Тому LPV Rate є точнішим показником для оцінки якості трафіку. Формула [1]:

$$LPV Rate = \frac{Landing\ Page\ Views}{Clicks} \cdot 100\% \quad (1.21)$$

Cost per LPV (вартість одного перегляду цільової сторінки) – це ціна, витрачена на отримання одного реального перегляду сторінки. Формула:

$$Cost\ per\ LPV = \frac{Ad\ Spend}{Landing\ Page\ Views} \quad (1.22)$$

1.2.5 Метрики для кампаній «Охоплення»

Для кампаній типу «Охоплення» головною метою є охоплення якомога більшої кількості унікальних користувачів. Тому важливими є Reach, Frequency, Reach Rate, CPM та Cost per 1000 Reach.

Cost per 1000 Reach – показує, скільки коштує охоплення тисячі унікальних користувачів. Розраховується за формулою:

$$Cost\ per\ 1000\ Reach = \frac{Ad\ Spend}{Reach} \cdot 1000 \quad (1.23)$$

1.2.6 Метрики для кампаній «Залучення»

Для кампаній типу «Залучення» важливо оцінити, як користувачі взаємодіють із рекламним контентом: ставлять лайки, коментують, поширюють або зберігають публікацію.

Engagement Rate – це відсоток взаємодій від кількості показів. Розраховується за формулою:

$$Engagement\ Rate = \frac{Engagements}{Impressions} \cdot 100\% \quad (1.24)$$

Cost per Engagement – це вартість однієї взаємодії. Розраховується за формулою:

$$Cost\ per\ Engagement = \frac{Ad\ Spend}{Engagements} \quad (1.25)$$

Окремо обчислюються детальні показники для кожного типу взаємодії : Comment Rate, Share Rate, Save Rate. Усі вони мають однакову структуру формули:

$$Comment\ Rate = \frac{Comments}{Impressions} \cdot 100\% \quad (1.26)$$

$$\text{Share Rate} = \frac{\text{Shares}}{\text{Impressions}} \cdot 100\% \quad (1.27)$$

$$\text{Save Rate} = \frac{\text{Saves}}{\text{Impressions}} \cdot 100\% \quad (1.28)$$

1.3 Benchmark-показники як орієнтир для оцінювання рекламних кампаній

Під час аналізу рекламної кампанії важливо не лише розрахувати її показники, а й зрозуміти, як їх правильно оцінювати. Наприклад, CTR, CPC, CPA або ROAS показують окремі сторони роботи кампанії, але не завжди дають повну картину без урахування мети кампанії, її типу, маржинальності, витрат і загального результату. Benchmark-показники в такому випадку можуть використовуватися як допоміжний орієнтир, який допомагає порівняти фактичні значення з типовими або середніми значеннями для подібних кампаній.

Benchmark – це орієнтовне або середнє значення певної рекламної метрики, яке використовується для порівняння з фактичними результатами кампанії. Він не є абсолютним критерієм успішності, але допомагає краще зрозуміти, наскільки отримані значення відрізняються від типових або очікуваних показників.

У цифровій рекламі benchmark-показники можуть використовуватися для різних метрик: CTR, CPC, CPM, CPA, CPL, ROAS, Conversion Rate та інших. Наприклад, якщо фактичний CPC кампанії значно вищий за benchmark, це може вказувати на високу вартість залучення користувачів. Якщо CTR нижчий за benchmark, це може бути сигналом, що рекламне оголошення недостатньо привертає увагу аудиторії. Водночас такі висновки потрібно перевіряти разом з іншими метриками, оскільки окремих показник не завжди відображає реальну ефективність кампанії.

Для порівняння результатів рекламної кампанії можуть використовуватися два типи орієнтирів:

— Зовнішні benchmark-показники, тобто середні значення, отримані з відкритих аналітичних звітів, рекламних платформ або досліджень. Вони можуть бути загальними або розрахованими для конкретної галузі.

— Внутрішні середні значення, тобто власна база порівняння рекламодавця. Вона корисна тоді, коли потрібно порівняти кілька кампаній між собою або оцінити зміну результатів за різні періоди.

У розробленому застосунку використовуються зовнішні benchmark-показники з відкритих аналітичних джерел. Для кампаній типу Sales одним із джерел є аналітичний матеріал Triple Whale щодо benchmark-показників Meta Ads, оновлений у 2026 році, де розглядаються CPA, CPM, ROAS, CTR та інші показники [2]. Для кампаній типу Lead Generation і частково Traffic використовуються дані WordStream / LocaliQ 2025, зокрема для CTR, CPC, CPL та Conversion Rate [3]. Також у застосунку передбачено поділ benchmark-показників на загальні та галузеві. Галузеві значення використовуються для таких сфер, як мода, краса та догляд, електроніка, товари для дому та саду, харчові продукти й напої, а також автомобільна сфера. Якщо для вибраної галузі немає окремого значення певної метрики, застосунок може використовувати загальний benchmark як резервний орієнтир.

Важливо враховувати, що benchmark не враховує всіх особливостей конкретного бізнесу: ціну товару, маржинальність, якість аудиторії, сезонність, рекламну стратегію та інші фактори. Наприклад, кампанія може мати CPA вищий за ринковий benchmark, але залишатися прибутковою завдяки високому середньому чеку або великій маржі.

Тому benchmark-показники в цій роботі розглядаються як допоміжний інструмент для аналізу, а не як остаточна оцінка успішності кампанії. Вони допомагають помітити можливі сильні та слабкі місця, але фінальний висновок

має формуватися з урахуванням типу кампанії, її мети, економічних показників і загальної динаміки результатів.

1.4 Аналіз існуючих рішень для рекламної аналітики

На сьогодні існує багато інструментів, які допомагають рекламодавцям аналізувати результати рекламних кампаній. Їх можна умовно поділити на кілька груп: рекламні кабінети, сервіси веб-аналітики, ВІ-інструменти для побудови звітів та спеціалізовані маркетингові платформи. Кожен із цих інструментів має свої переваги, але водночас не завжди повністю закриває потребу у простому та зрозумілому аналізі рекламних даних.

Першу групу становлять рекламні кабінети самих платформ, наприклад Meta Ads Manager, Google Ads та TikTok Ads Manager. Їхньою основною перевагою є прямий доступ до даних кампаній. Користувач може переглядати покази, кліки, витрати, охоплення, конверсії та інші показники без додаткового імпорту даних. Такі інструменти зручні для запуску реклами, щоденного контролю кампаній, зміни бюджету та перегляду базової статистики.

Водночас рекламні кабінети зазвичай працюють тільки в межах власної платформи. Якщо рекламодавець використовує кілька каналів просування, наприклад Meta Ads, Google Ads і TikTok Ads, йому потрібно переглядати дані окремо в кожній системі або вручну об'єднувати їх у таблицях. Крім того, такі кабінети переважно показують числові значення, але не завжди пояснюють, чому певний результат можна вважати сильним або проблемним. Через це користувач все одно має самостійно аналізувати метрики, порівнювати їх між собою та робити висновки. Також варто зазначити, що в рекламних інструментах Meta, активно використовуються функції штучного інтелекту й автоматизації

Advantage+, які можуть автоматично оптимізувати аудиторії, плейсменти, бюджет і креативи рекламної кампанії. За інформацією Meta, можливості Advantage+ поступово розширюються, а частина нових функцій проходить тестування або впроваджується поступово [4]. Тому автоматичні рекомендації рекламного кабінету не завжди варто сприймати як остаточну оцінку ефективності кампанії. Вони можуть бути корисними як підказка, але не завжди враховують специфіку конкретного бізнесу, маржинальність товарів, якість заявок або фактичний прибуток.

Другу групу становлять сервіси веб-аналітики, наприклад Google Analytics. Вони надають багато даних про поведінку користувачів після переходу на сайт: перегляди сторінок, джерела трафіку, події, конверсії, шляхи користувачів та інші дії. Такі інструменти корисні для аналізу якості трафіку та оцінювання того, як користувачі взаємодіють із сайтом після переходу з реклами. Водночас Google Analytics більше орієнтований саме на веб-аналітику, тому для окремого аналізу рекламних кампаній часто потрібно додатково враховувати рекламні витрати, прибутковість, benchmark-показники, тип кампанії та порівняння з іншими кампаніями або періодами.

Третю групу складають BI-інструменти та системи побудови звітів, наприклад Looker Studio або Power BI. Вони дозволяють створювати дашборди, підключати різні джерела даних і будувати візуальні звіти. Але така частина інструментів частіше платні, що не завжди зручно, якщо потрібно швидко перевірити окрему кампанію без повноцінного налаштування аналітичної системи.

Окремим простим способом аналізу залишаються електронні таблиці, наприклад Microsoft Excel або Google Sheets. Вони дозволяють вручну вносити дані, створювати формули та будувати базові графіки. Перевагою такого підходу є доступність і гнучкість. Проте ручна робота з таблицями може призводити до помилок у формулах, некоректного введення даних або складнощів під час

порівняння кількох кампаній і періодів. Також таблиці не дають рекомендації щодо рекламних кампаній.

Порівняння існуючих рішень показує, що більшість із них закриває лише частину задач рекламної аналітики. Рекламні кабінети зручні для перегляду статистики в межах однієї платформи, але не завжди дають можливість швидко порівнювати кампанії, періоди, benchmark-показники та формувати пояснювальні рекомендації. ВІ-інструменти і таблиці дозволяють створювати власні звіти, однак потребують додаткового налаштування формул, графіків і правил аналізу. Тому в межах дипломної роботи передбачено створення веб-застосунку, який поєднує введення або завантаження даних, автоматичний розрахунок рекламних метрик, порівняння кампаній і періодів, benchmark-аналіз, z-score-аналіз, кореляційний аналіз, прогнозування та формування рекомендацій в одному інструменті.

1.5 Постановка задачі та вимоги до веб-застосунку

На основі огляду предметної області, аналізу основних рекламних метрик, benchmark-показників та існуючих рішень можна сформулювати задачу до веб-застосунка. Необхідно розробити веб-застосунок для аналізу та прогнозування ефективності рекламних кампаній, який дозволяє користувачу вводити рекламні дані, автоматично обчислювати ключові показники, порівнювати результати та отримувати рекомендації щодо покращення рекламних кампаній.

У межах дипломної роботи веб-застосунок має забезпечувати:

— введення даних рекламної кампанії вручну або завантаження даних із CSV-файлу;

- аналіз однієї кампанії, порівняння періодів та порівняння кількох кампаній між собою;
- автоматичний розрахунок ключових рекламних показників;
- порівняння результатів із benchmark-показниками, між кампаніями та між окремими періодами;
- виявлення відхилень за допомогою z-score-аналізу;
- аналіз взаємозв'язків між показниками за допомогою кореляційного аналізу;
- прогнозування результатів рекламної кампанії через рекламну воронку та лінійну регресію;
- формування рекомендацій для покращення рекламних результатів;
- візуалізацію результатів у вигляді таблиць, карток показників і графіків.

Також він повинен мати зрозумілий інтерфейс, працювати у браузері, коректно обробляти неповні або помилкові вхідні дані, а також виконувати розрахунки без передавання рекламної інформації на сторонні сервери.

Таким чином, розроблений веб-застосунок має поєднувати розрахунок рекламних метрик, статистичний аналіз, порівняння, прогнозування та формування рекомендацій в одному інструменті, зручному для швидкого аналізу рекламних кампаній.

РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ АНАЛІЗУ РЕКЛАМНИХ КАМПАНІЙ

Під час аналізу рекламних кампаній важливо не тільки розрахувати основні метрики, а й правильно їх оцінити. Потрібно зрозуміти, чи є серед показників незвичні значення, які сильно відрізняються від інших, а також чи пов'язані між собою окремі метрики. Для цього в роботі використовуються z-score-аналіз та кореляція Пірсона.

2.1 Z-score для виявлення відхилень

Z-score використовується для визначення того, наскільки конкретне значення показника відрізняється від середнього рівня. Перед його розрахунком визначаються два допоміжні статистичні показники: середнє значення та стандартне відхилення.

Середнє значення показує, яким є звичайний рівень певної метрики в наборі даних. Наприклад, якщо аналізується кілька кампаній або періодів, можна знайти середній CTR, CPC чи ROAS і порівняти з ним результат окремої кампанії. Розраховується за формулою [5]:

$$\underline{x} = \frac{1}{n} \times \sum_{i=1}^n x_i, \quad (2.1)$$

де x_i – окреме значення показника;

n – кількість спостережень.

Стандартне відхилення показує, наскільки сильно окремі значення відрізняються від середнього. Якщо стандартне відхилення невелике, значення

розташовані близько одне до одного. Якщо воно велике, між кампаніями, періодами або днями є суттєва різниця. Розраховується за формулою:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}, \quad (2.2)$$

де σ – стандартне відхилення;

x_i – окреме значення показника;

\bar{x} – середнє значення показника;

n – кількість спостережень.

Після цього розраховується Z-score:

$$Z = \frac{x - \bar{x}}{\sigma}, \quad (2.3)$$

де x – значення показника.

Інтерпретація Z-score залежить від абсолютного значення $|Z|$:

— $|Z| < 1$ – показник вважається близьким до середнього рівня;

— $1 \leq |Z| < 2$ – це означає помірне відхилення від середнього;

— $|Z| \geq 2$ – відхилення вважається значним і потребує додаткової уваги

під час аналізу [5].

Під час інтерпретації Z-score важливо враховувати напрям метрики, тобто чи є збільшення показника позитивним або негативним сигналом. Сам по собі Z-score лише показує, наскільки значення відхиляється від середнього рівня, але не завжди означає добрий або поганий результат.

До метрик, для яких вище значення зазвичай є кращим, належать CTR, CR/CVR, Lead CR, LPV Rate, Engagement Rate, Reach Rate, ROAS, Profit, Gross Profit, Gross Margin, Revenue, Purchases, Leads, Qualified Leads, Landing Page Views, Engagements, Comments, Shares, Saves, Reach та AOV. Для цих показників

додатний z-score може свідчити про кращий результат порівняно із середнім рівнем, а від'ємний z-score про слабший результат.

До метрик, для яких нижче значення зазвичай є кращим, належать CPC, CPM, CPA, CPL, Cost per LPV, Cost per Engagement, Cost per 1000 Reach та Cost per Qualified Lead. Для цих показників додатний z-score може бути негативним сигналом, оскільки вартість або витрати вищі за середній рівень. Від'ємний z-score для таких метрик, навпаки, може бути позитивним, оскільки означає нижчу вартість порівняно із середнім значенням.

Для метрики Frequency не завжди діє правило «більше – краще» або «менше – краще», оскільки занадто низька частота може означати недостатню кількість контактів з аудиторією, а занадто висока що аудиторія перенасичена рекламою. Тому відхилення Frequency потрібно оцінювати разом із CTR, CPM, Reach та загальною метою кампанії.

2.2 Кореляція Пірсона для аналізу зв'язків між показниками

Під час аналізу рекламних кампаній важливо не лише оцінювати окремі метрики, а й розуміти, як вони можуть бути пов'язані між собою. Наприклад, збільшення бюджету може супроводжуватися зростанням кількості покупок, а підвищення частоти показів може збігатися зі зниженням клікабельності реклами. Для виявлення таких зв'язків у роботі використовується кореляція Пірсона.

Кореляція Пірсона – це статистичний метод, який дозволяє оцінити силу та напрям лінійного зв'язку між двома числовими показниками. Коефіцієнт кореляції позначається (r) і набуває значень від (-1) до (1). Формула має вигляд:

$$r = \frac{\text{cov}(x,y)}{\sigma_x \cdot \sigma_y}, \quad (2.4)$$

де $\text{cov}(x, y)$ – коваріація між двома показниками;

σ_x, σ_y стандартні відхилення цих показників.

Формула коваріації між двома показниками:

$$\text{cov}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}), \quad (2.5)$$

де x_i та y_i – окремі значення двох показників;

\bar{x} та \bar{y} – їхні середні значення;

n – кількість спостережень.

Після розрахунку коефіцієнта кореляції його значення показує напрям зв'язку між рекламними метриками. Якщо (r) близький до (1), показники зазвичай зростають разом. Наприклад, зі збільшенням Ad Spend може зростати кількість Purchases або Leads. Якщо (r) близький до (-1), зі зростанням одного показника інший зазвичай зменшується. Наприклад, збільшення Frequency може супроводжуватися зниженням CTR. Якщо (r) близький до (0), помітного лінійного зв'язку між метриками не виявлено.

Для інтерпретації сили зв'язку використовується абсолютне значення $|r|$. Якщо ($|r| < 0,3$), зв'язок вважається слабким. Якщо ($0,3 \leq |r| < 0,7$), зв'язок є помірним. Якщо ($|r| \geq 0,7$), зв'язок вважається сильним.

У розробленому застосунку кореляційний аналіз виконується лише за наявності щонайменше трьох коректних пар значень. Якщо даних менше або один із показників не має числового значення, кореляція не розраховується. Це потрібно для того, щоб не робити висновки на основі занадто малого набору даних.

Кореляційний аналіз може виконуватися в різних режимах залежно від доступних даних. Якщо користувач завантажує щоденні дані кампанії, система аналізує зв'язки між показниками в динаміці за днями. Якщо порівнюється кілька кампаній або кілька періодів, кореляція може розраховуватися між їхніми агрегованими значеннями, але лише за умови достатньої кількості спостережень.

Набір пар показників для кореляційного аналізу залежить від типу рекламної кампанії. Наприклад, для кампаній типу «Продажі» аналізуються зв'язки між Frequency і CTR, CPC і CPA, CR і CPA, ROAS і Profit, Ad Spend і Purchases. Такі пари допомагають зрозуміти, чи пов'язана частота показів із клікабельністю, чи впливає вартість кліку на вартість покупки, а також чи супроводжується збільшення бюджету зростанням кількості покупок.

Для кампаній типу «Лідогенерація» важливими є зв'язки між Frequency і CTR, CTR і CPL, Lead CR і CPL, Qualified Lead Rate і Cost per Qualified Lead, Ad Spend і Leads. У цьому випадку кореляційний аналіз допомагає оцінити, чи впливає якість трафіку на вартість заявки та чи збільшення бюджету супроводжується збільшенням кількості лідів.

Для кампаній типу «Трафік» аналізуються пари CTR і CPC, CPM і CPC, LPV Rate і Cost per LPV, Frequency і CTR, Ad Spend і Landing Page Views. Для кампаній типу «Охоплення» важливими є зв'язки між Ad Spend і Reach, Frequency і Reach, CPM і Cost per 1000 Reach, Frequency і Cost per 1000 Reach. Для «Залучення» аналізуються Frequency і Engagement Rate, CPM і Cost per Engagement, Engagement Rate і Cost per Engagement, Ad Spend і Engagements, CTR і Engagement Rate.

Важливо враховувати, що кореляція не доводить причинно-наслідковий зв'язок. Наприклад, якщо між Ad Spend і Purchases є позитивна кореляція, це не означає, що збільшення бюджету завжди автоматично призведе до зростання покупок. На результат можуть впливати сезонність, якість креативу, аудиторія,

ціна товару, конкуренція та інші фактори. Тому кореляція в застосунку використовується як аналітична підказка, а не як остаточний доказ причини.

2.3 Методи прогнозування результатів рекламної кампанії

У розробленому застосунку прогнозування використовується для оцінки можливих результатів рекламної кампанії при зміні бюджету. Користувач задає новий денний бюджет, кількість днів прогнозу та обирає метод прогнозування. Після цього система розраховує очікувані результати залежно від типу кампанії. Передбачено два основні підходи до прогнозування: прогноз через рекламну воронку та прогноз за допомогою лінійної регресії.

2.3.1 Прогноз через рекламну воронку

Прогноз через рекламну воронку базується на послідовному переході від рекламного бюджету до кінцевого результату кампанії. Загальна логіка такого прогнозу має вигляд:

бюджет → покази → кліки/охоплення → цільова дія

Спочатку розраховується загальна сума витрат на прогнозований період:

$$TotalSpend = DailyBudget \cdot ForecastDays, \quad (2.6)$$

де *DailyBudget* – новий денний бюджет;

ForecastDays – кількість днів прогнозу.

Для кампаній типу «Продажі» прогноз розраховується за такою послідовністю:

$$1. \quad Impressions = \frac{TotalSpend}{CPM} \cdot 1000, \quad (2.7)$$

$$2. \quad Clicks = Impressions \cdot \frac{CTR}{100}, \quad (2.8)$$

$$3. \quad Purchases = Clicks \cdot \frac{CR}{100}, \quad (2.9)$$

$$4. \quad Revenue = Purchases \cdot AOV, \quad (2.10)$$

$$5. \quad Gross Profit = Revenue \cdot \frac{GrossMargin}{100}, \quad (2.11)$$

$$6. \quad Profit = Gross Profit - TotalSpend \quad (2.12)$$

$$7. \quad ROAS = \frac{Revenue}{TotalSpend} \quad (2.13)$$

$$8. \quad CPA = \frac{TotalSpend}{Purchases} \quad (2.14)$$

Для кампаній типу «Лідогенерація» прогноз має таку послідовність:

$$1. \quad Impressions = \frac{TotalSpend}{CPM} \cdot 1000,$$

$$2. \quad Clicks = Impressions \cdot \frac{CTR}{100},$$

$$3. \quad Leads = Clicks \cdot \frac{LeadCR}{100}, \quad (2.15)$$

$$4. \quad CPL = \frac{TotalSpend}{Leads}, \quad (2.16)$$

Для кампаній типу «Трафік» прогноз розраховується так:

$$1. \quad Impressions = \frac{TotalSpend}{CPM} \cdot 1000,$$

$$2. \quad Clicks = Impressions \cdot \frac{CTR}{100},$$

$$3. \quad LandingPageViews = Clicks \cdot \frac{LPVRate}{100}, \quad (2.19)$$

$$4. \quad CPC = \frac{TotalSpend}{Clicks}, \quad (2.20)$$

$$5. \quad CostPerLPV = \frac{TotalSpend}{LandingPageViews}, \quad (2.21)$$

Для «Охоплення» використовується така послідовність:

$$1. \quad Impressions = \frac{TotalSpend}{CPM} \cdot 1000,$$

$$2. \quad Reach = \frac{Impressions}{Frequency} \quad (2.22)$$

$$3. \quad CostPer1000Reach = \frac{TotalSpend}{Reach}, \quad (2.24)$$

Для «Залучення» прогноз розраховується так:

$$1. \quad Impressions = \frac{TotalSpend}{CPM} \cdot 1000,$$

$$2. \quad Engagements = Impressions \cdot \frac{EngagementRate}{100} \quad (2.25)$$

$$3. \quad CostPerEngagement = \frac{TotalSpend}{Engagements} \quad (2.26)$$

Воронковий метод прогнозування є простим і інтуїтивно зрозумілим, проте він має певне обмеження: він припускає, що між бюджетом і результатом зберігається приблизно пропорційна залежність. У реальних рекламних кампаніях ця залежність не завжди є сталою. При значному збільшенні бюджету ефективність може знижуватися через насичення аудиторії, зростання конкуренції в рекламному аукціоні або погіршення якості трафіку при розширенні таргетингу. Тому воронковий метод доцільно використовувати для орієнтовного прогнозу та помірних змін бюджету, особливо тоді, коли доступні

дані лише за одну кампанію або один період. Якщо є історичні дані за кілька днів, періодів або кампаній, більш обґрунтованим може бути використання регресійного прогнозування.

2.3.2 Прогноз через лінійну регресію

Лінійна регресія використовується для прогнозування результатів рекламної кампанії на основі історичних даних. На відміну від воронкового методу, який використовує поточні коефіцієнти кампанії, регресія аналізує фактичний зв'язок між рекламними витратами та результатом у попередніх спостереженнях.

Лінійна регресія описується формулою:

$$y = a \cdot x + b, \quad (2.27)$$

де y – прогнозований показник;

x – рекламні витрати;

a – коефіцієнт нахилу, який показує, як змінюється результат при зміні витрат;

b – базове значення моделі.

Для обчислення коефіцієнтів a і b використовується метод найменших квадратів. Спочатку обчислюються середні значення \bar{x} та \bar{y} за формулою (2.1). Коефіцієнт нахилу a розраховується за формулою:

$$a = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.28)$$

Після цього обчислюється вільний член b :

$$b = \bar{y} - a \cdot \bar{x} \quad (2.29)$$

Після знаходження коефіцієнтів a і b система підставляє у формулу нове значення бюджету x_{new} , задане користувачем, і отримує прогнозоване значення показника:

$$\hat{y} = a \cdot x_{new} + b, \quad (2.30)$$

де \hat{y} – прогнозоване значення показника.

У застосунку лінійна регресія використовується за наявності достатньої кількості коректних спостережень. Мінімально для побудови прямої потрібно два спостереження, але для практичного прогнозування бажано мати більше даних. Якщо даних недостатньо або всі значення x однакові, регресійну модель побудувати неможливо. Після прогнозування основних результатів система додатково розраховує похідні показники. Наприклад, якщо користувач задає прогнозний бюджет, а система прогнозує кількість кліків, покупок і дохід, то на основі цих значень можна розрахувати прогнозні CPC, CPA, ROAS, CTR, CR та інші метрики. Завдяки цьому користувач бачить не тільки очікувану кількість результатів, а й майбутню ефективність кампанії.

2.3.3 Сценарний прогноз

Щоб прогноз був більш корисним для користувача, прогноз формується у трьох сценаріях: базовому, оптимістичному та песимістичному. Базовий сценарій використовує фактичні коефіцієнти поточної кампанії без змін:

$$k_{base} = k, \quad (2.31)$$

де k – поточне значення показника або коефіцієнта рекламної воронки.

Для воронкового прогнозу оптимістичний і песимістичний сценарії формуються через умовну зміну ключових коефіцієнтів. Це не є універсальним правилом ринку, а використовується як сценарне припущення для оцінки можливого діапазону результатів. В оптимістичному сценарії конверсійні показники збільшуються на 10%:

$$k_{opt} = k \cdot 1.1 \quad (2.32)$$

Це стосується таких показників, як CTR, CR, Lead CR, LPV Rate, Engagement Rate та Qualified Lead Rate. Для середнього чека кампанії Продажі використовується помірніше збільшення:

$$AOV_{opt} = AOV \cdot 1.05 \quad (2.33)$$

Вартість показів в оптимістичному сценарії зменшується:

$$CPM_{opt} = CPM \cdot 0.95 \quad (2.33)$$

У песимістичному сценарії конверсійні показники зменшуються на 10%, середній чек для Продажі зменшується на 5%, а вартість показів зростає:

$$k_{pes} = k \cdot 0.9 \quad (2.34)$$

$$AOV_{pes} = AOV \cdot 0.95 \quad (2.35)$$

$$CPM_{pes} = CPM \cdot 1.1 \quad (2.36)$$

Така асиметрія для CPM враховує те, що в рекламному аукціоні зростання вартості показів може сильніше впливати на результат кампанії, ніж її невелике зниження. Після зміни коефіцієнтів система повторно виконує розрахунок прогнозу за формулами рекламної воронки.

У застосунку також виконується нормалізація значень: відсоткові показники обмежуються діапазоном від 0 до 100, а вартісні показники не можуть бути від'ємними.

Для регресійного прогнозу сценарії формуються інакше. Спочатку система будує базову лінійну модель:

$$\widehat{y_{base}} = a \cdot x_{new} + b \quad (2.37)$$

де x_{new} – новий бюджет, заданий користувачем, а $\widehat{y_{base}}$ – базове прогнозне значення. Після цього розраховується залишкова стандартна похибка моделі:

$$RSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \widehat{y}_i)^2}{n-2}} \quad (2.38)$$

де y_i – фактичне значення показника;

\widehat{y}_i – значення, розраховане регресійною моделлю;

n – кількість спостережень.

Такий розрахунок похибки застосовується, якщо кількість спостережень більша за два. На основі RSE формується оцінка можливого відхилення прогнозу:

$$\hat{y}_{opt} = \hat{y}_{base} + RSE \quad (2.39)$$

$$\hat{y}_{pes} = \hat{y}_{base} - RSE \quad (2.40)$$

Ці сценарії не є точним довірчим інтервалом, але дозволяють показати приблизний діапазон можливих результатів. Якщо прогнозований показник не може бути від'ємним, система обмежує його мінімальне значення нулем. Після цього похідні метрики, такі як CPC, CPA, ROAS, CTR або CR, розраховуються повторно вже на основі прогнозних значень.

2.4 Методика комплексної оцінки Campaign Score

Для узагальнення результатів аналізу в застосунку використовується комплексна оцінка Campaign Score. Це інтегральний показник у діапазоні від 0 до 100, який дозволяє швидко оцінити загальний стан рекламної кампанії. Він не замінює детальний аналіз окремих метрик, але допомагає користувачу одразу побачити, чи є кампанія сильною, середньою або проблемною.

Campaign Score потрібен тому, що рекламна кампанія не може оцінюватися лише за одним показником. Наприклад, кампанія може мати високий CTR, але низьку конверсію, або дешеві ліди, але слабку якість заявок. Тому в оцінці враховується кілька ключових метрик одночасно.

2.4.1 Загальна логіка розрахунку

Для кожної метрики спочатку розраховується часткова оцінка від 0 до 100. Якщо для метрики більше значення є кращим, використовується формула:

$$Score_i = \min\left(\frac{Actual_i}{Target_i} \cdot 100, 100\right) \quad (2.41)$$

Якщо для метрики менше значення є кращим, використовується обернена формула:

$$Score_i = \min\left(\frac{Target_i}{Actual_i} \cdot 100, 100\right) \quad (2.42)$$

де $Actual_i$ – фактичне значення показника;

$Target_i$ – орієнтир для порівняння.

Таким орієнтиром може бути зовнішній benchmark або внутрішнє середнє значення, якщо користувач аналізує кілька кампаній чи періодів.

Після цього часткові оцінки об'єднуються за допомогою вагових коефіцієнтів:

$$CampaignScore = \sum_{i=1}^n Score_i \cdot w_i, \quad (2.43)$$

де w_i – вага окремої метрики. Якщо частина метрик недоступна, вона не враховується, а ваги доступних метрик перераховуються між собою.

2.4.2 Ваги метрик

У Campaign Score різні типи рекламних кампаній оцінюються за різними наборами метрик, оскільки їхні цілі відрізняються. Вагові коефіцієнти в Campaign Score задані як авторська експертна евристика на основі цілей різних типів рекламних кампаній. Вони не є універсальним галузевим стандартом, а використовуються для того, щоб у межах розробленого застосунку надати більшу вагу тим метрикам, які найбільше відповідають меті конкретного типу кампанії. Для кампаній з продажів найбільше значення мають економічні показники, для лідогенерації це вартість і якість заявок, для трафіку – ефективність переходів на цільову сторінку, для охоплення – вартість охоплення аудиторії, а для залучення – активність користувачів щодо рекламного контенту.

У таблиці 2.1 наведено ваги метрик, які використовуються для розрахунку Campaign Score у застосунку.

Таблиця 2.1 — Ваги метрик для розрахунку Campaign Score

Тип кампанії	Основна схема ваг	Альтернативна схема ваг
Продажі	Profit – 40%; ROAS – 25%; CPA – 20%; CTR – 7,5%; CR – 7,5%	Якщо Profit недоступний: ROAS – 50%; CPA – 35%; CTR – 7,5%; CR – 7,5%
Лідогенерація	Generation CPL – 35%; Lead CR – 25%; Qualified Lead Rate – 25%; CTR – 15%	Якщо Qualified Lead Rate недоступний: CPL – 45%; Lead CR – 35%; CTR – 20%
Трафік	CPC – 30%; LPV Rate – 30%; CTR – 25%; CPM – 15%	Не використовується

Охоплення	Cost per 1000 Reach – 35%; Reach Rate – 25%; CPM – 25%; Frequency – 15%	Якщо Reach Rate недоступний: Cost per 1000 Reach – 45%; CPM – 35%; Frequency – 20%
Залучення	Engagement Rate – 40%; Cost per Engagement – 30%; CTR – 15%; Frequency – 15%	Не використовується

2.4.3 Особливості оцінювання Sales-кампаній

Для кампаній типу Продажі частина оцінки будується на економічній логіці кампанії. У цьому випадку важливо враховувати не лише benchmark-показники, а й те, чи є кампанія прибутковою. ROAS порівнюється з Break-even ROAS, тобто з рівнем безбитковості. CPA порівнюється з Max Profitable CPA, тобто з максимально допустимою вартістю покупки, за якої кампанія ще може залишатися прибутковою.

Profit оцінюється окремо через співвідношення прибутку до рекламних витрат:

$$ProfitRatio = \frac{Profit}{Ad Spend} \quad (2.44)$$

Якщо прибуток є високим відносно рекламних витрат, часткова оцінка за цією метрикою наближається до 100. Якщо прибуток від'ємний, оцінка знижується, оскільки кампанія не покриває витрати та приносить збиток.

2.4.4 Корекція змішаних сигналів

У застосунку передбачено корекцію ситуацій, коли одна дуже сильна метрика може приховати слабкість інших показників. Наприклад, кампанія може мати дуже хороший CPL, але слабкі CTR і Lead CR. У такому випадку проста середньозважена оцінка могла б бути завищеною.

Для цього система визначає слабкі та сильні метрики. Слабкими вважаються часткові оцінки нижче 85, а сильними оцінки від 95 і вище. Якщо одночасно є сильні й слабкі сигнали, загальна оцінка коригується в бік слабших показників.

Також при порівнянні періодів Campaign Score додатково коригується залежно від динаміки. Якщо поточний період покращився відносно попереднього, оцінка збільшується на 3 бали, але не може перевищувати 100. Якщо динаміка змішана, оцінка обмежується значенням 92. Якщо поточний період погіршився, оцінка зменшується на 6 балів. Додатково для погіршення застосовується верхнє обмеження: якщо негативних показників два або більше, оцінка не може бути вищою за 82; якщо негативний показник один, оцінка не може бути вищою за 88.

2.4.5 Корекція змішаних сигналів

У застосунку Campaign Score інтерпретується так як наведено в таблиці 2.2:

Таблиця 2.2 — Інтерпретація Campaign Score

Значення Campaign Score	Інтерпретація
80-100	сильна кампанія
60-79	середня кампанія
40-59	слабка кампанія
0-39	критична кампанія

У застосунку для сильних кампаній також можуть відобразитися уточнення, наприклад наявність окремих зон оптимізації або негативної динаміки при порівнянні періодів.

2.5 Формування рекомендацій на основі результатів аналізу

Після розрахунку рекламних метрик застосунок формує рекомендації, які допомагають користувачу зрозуміти, які саме показники кампанії потребують уваги. Рекомендації не є випадковими підказками, а будуються на логіці роботи цифрової реклами: аналізі цілі кампанії, ефективності витрат, якості трафіку, конверсій, охоплення, залучення аудиторії та динаміки показників.

Подібний підхід використовується і в рекламних платформах. У Google Ads рекомендації допомагають покращувати кампанії через пропозиції щодо ставок, бюджету, ключових слів, оголошень та інших елементів, а також можуть враховувати історичні дані й актуальні тенденції [6]. У Meta Ads Manager використовується Opportunity Score, який пов'язаний із рекомендаціями для кампаній, груп оголошень та оголошень. При цьому Meta зазначає, що

застосування таких рекомендацій може допомогти покращити результати, але не гарантує фактичну ефективність кампанії [4].

У застосунку рекомендації будуються на основі фактичних метрик, benchmark-показників або внутрішніх середніх значень, z-score-аналізу, кореляційного аналізу та порівняння динаміки між періодами. Застосунок враховує тип рекламної кампанії та напрям ключових метрик. Це дозволяє оцінювати кампанію не за одним універсальним показником, а відповідно до її цілі: продажів, лідогенерації, трафіку, охоплення або залучення.

Z-score-аналіз та кореляційний аналіз у цьому блоці використовуються як додаткові сигнали для рекомендацій. Якщо z-score показує значне відхилення метрики, застосунок радить перевірити можливу причину такого стрибка або падіння. Якщо кореляційний аналіз виявляє сильний зв'язок між показниками, система використовує його як підказку для пояснення проблеми, наприклад можливої втрати аудиторії при зростанні Frequency і падінні CTR.

У застосунку рекомендації залежать від типу кампанії та виявлених проблем. Наприклад, для кампаній типу Продажі система може рекомендувати поступове збільшення бюджету, якщо кампанія є прибутковою, або навпаки радити не масштабувати її, якщо CPA перевищує допустимий рівень чи кампанія є збитковою. Для Лідогенерації рекомендації можуть стосуватися не лише вартості ліда, а й його якості. Для кампаній типу Трафік система може звертати увагу на якість переходів на цільову сторінку, для Охоплення – на охоплення та частоту показів, а для Залучення на рівень взаємодії з контентом.

Кожна рекомендація в інтерфейсі має однакову логіку подання: аналітичний висновок або проблема, підтверджувальні сигнали та рекомендована дія. Детальні приклади роботи рекомендаційного блоку наведено у розділі 4 Тестування та аналіз результатів.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ

3.1 Вибір технологій та середовища розробки

Для реалізації програмного продукту було обрано формат веб-застосунку, оскільки такий підхід дозволяє користувачу працювати з програмою через браузер без встановлення окремого десктопного програмного забезпечення. Користувач вводить або завантажує дані рекламних кампаній, після чого система виконує розрахунки, будує графіки, формує прогноз і рекомендації.

Застосунок реалізовано на Vanilla JavaScript , тобто без використання frontend-фреймворків React, Vue або Angular [7]. Такий підхід дозволив реалізувати власну модульну структуру застосунку та повністю контролювати логіку введення даних, розрахунку метрик, статистичного аналізу, прогнозування, формування рекомендацій і відображення інтерфейсу. Окремі частини логіки винесені в різні модулі: розрахунок рекламних метрик, робота з CSV-файлами, побудова графіків, прогнозування, Campaign Score, рекомендації та компоненти інтерфейсу.

Для створення сторінки застосунку використано HTML і CSS. HTML у проєкті використовується як єдина точка входу у застосунок: у файлі index.html визначено базову структуру сторінки, кореневий елемент #app і підключення основного JavaScript-модуля. CSS відповідає за зовнішній вигляд інтерфейсу, адаптивність, кольори, відступи, компоненти та коректне відображення сторінки на різних розмірах екрана.

Для запуску та збирання проєкту використовується Vite . Цей інструмент дозволяє швидко запускати локальний сервер під час розробки та формувати готову збірку застосунку [8]. Також у проєкті використовується vite-plugin-singlefile, завдяки якому фінальна версія може бути сформована як один HTML-файл із вбудованими JavaScript- і CSS-ресурсами. Це зручно для демонстрації

дипломного проекту, оскільки готовий файл можна відкрити у браузері без розгортання окремої серверної частини.

Для побудови графіків використовується бібліотека Chart.js. Вона застосовується для візуалізації результатів аналізу: порівняння фактичних значень із benchmark-показниками, відображення динаміки за днями, порівняння кампаній або періодів, а також показу сценаріїв прогнозу. Завдяки графікам користувач може оцінювати результати у візуальній формі.

Для роботи з CSV-файлами використовується бібліотека PapaParse. Вона дозволяє зчитувати табличні дані, які користувач завантажує у застосунок, і перетворювати їх у формат, придатний для подальшої обробки. Це важливо, оскільки рекламні кабінети часто дозволяють експортувати статистику саме у CSV-форматі [9].

Окремо слід зазначити, що застосунок не має серверної частини та бази даних. Усі основні розрахунки виконуються на стороні клієнта, тобто безпосередньо у браузері користувача. Node.js у проекті використовується не як backend-сервер, а як середовище для запуску інструментів розробки, зокрема Vite, ESLint і Prettier.

Узагальнений перелік використаних технологій наведено в таблиці 3.1.

Таблиця 3.1 — Технології, використані для розробки веб-застосунку

Технологія	Призначення
Vanilla JavaScript	Основна логіка застосунку, обробка даних, розрахунок метрик, аналіз і прогнозування;
HTML	єдина точка входу у застосунок
CSS	оформлення, адаптивність і візуальне представлення;
Vite	збірка проєкту;
Chart.js	JS бібліотека для побудови графіків в UI;
PapaParse	бібліотека для зчитування та обробки CSV-файлів.

3.2 Загальна структура веб-застосунку

Веб-застосунок має модульну структуру, тобто основні частини програми розділені за своїм призначенням. Це спрощує підтримку проєкту, оскільки розрахунки, робота з даними, інтерфейс, графіки, прогнозування та рекомендації не змішані в одному файлі, а винесені в окремі модулі (рис. 3.1).

Точкою входу у застосунок є файл `index.html`, у якому визначено кореневий елемент `#app` та підключено основний JavaScript-модуль `src/main.js`. Далі файл `main.js` завантажує стилі та запускає функцію `bootstrap`, яка створює

основну структуру інтерфейсу: бокове меню, заголовок, блок введення даних, блок зведених результатів, детальний аналіз, рекомендації, прогноз і графіки. Загальна структура проекту наведена в таблиці 3.2.

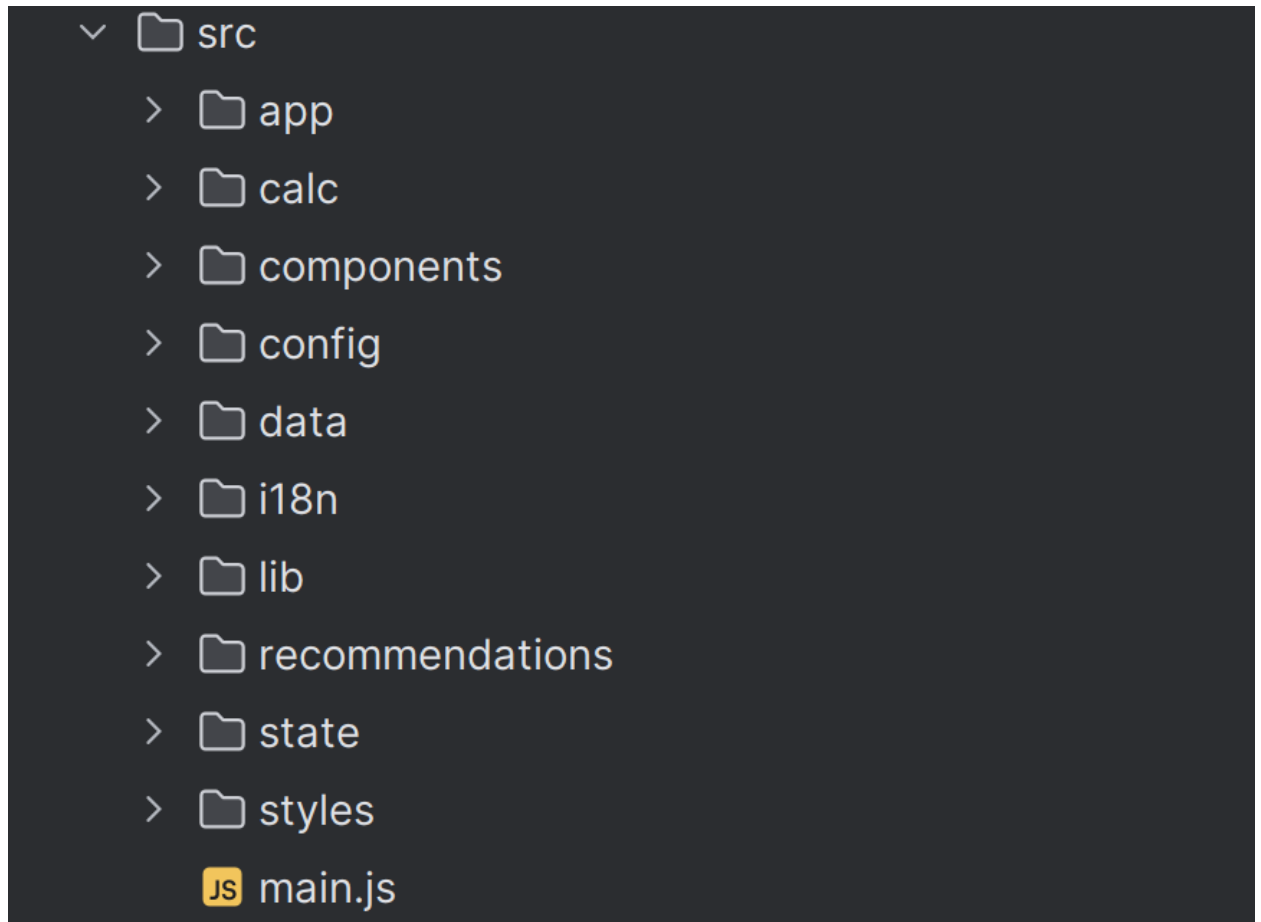


Рисунок 3.1 — Структура вихідного коду веб-застосунку

Таблиця 3.2 — Основні модулі веб-застосунку

Папка / модуль	Призначення
src/app	Запуск застосунку, створення основної структури сторінки та керування діями користувача
src/state	Зберігання поточного стану застосунку
src/components	Компоненти інтерфейсу: форми, таблиці, графіки, рекомендації, прогноз, меню
src/data	Обробка введених даних, CSV-файлів і нормалізація даних перед аналізом
src/calc	Розрахунок метрик, статистичних показників, Campaign Score, кореляцій, прогнозу та порівнянь
src/recommendations	Формування рекомендацій на основі результатів аналізу
src/config	Налаштування типів кампаній, полів, benchmark-показників, ваг метрик і сценаріїв прогнозу
src/lib	Допоміжні функції для DOM, форматування, валідації та іконок
src/i18n	Тексти інтерфейсу українською мовою
src/styles	CSS-файли для оформлення інтерфейсу

Логіка роботи застосунку починається з вибору типу аналізу та введення даних. Користувач може аналізувати одну кампанію, декілька кампаній або декілька періодів. Дані можуть вводитися вручну або завантажуватися з CSV-файлу. За введення даних відповідають компоненти з папки `src/components`, зокрема форма кампанії, блок вибору способу введення та компонент завантаження CSV.

Перед виконанням аналізу дані проходять перевірку та нормалізацію. Цей етап реалізовано у модулях `src/data` та `src/lib/validation.js`. Система перевіряє обов'язкові поля, числові значення, дати, тип кампанії та інші параметри. Після цього дані приводяться до формату, з яким можуть працювати розрахункові модулі.

Основні розрахунки виконуються в папці `src/calc`. Модуль `metrics.js` відповідає за обчислення рекламних метрик, `statistics.js` за статистичні методи, зокрема середнє значення, стандартне відхилення, *Z-score*, кореляцію Пірсона та лінійну регресію. Модуль `campaign-score.js` розраховує комплексну оцінку Campaign Score, `forecast.js` відповідає за прогнозування, а `correlations.js` за аналіз зв'язків між рекламними показниками. Загальне об'єднання результатів аналізу виконується в модулі `analysis.js`.

Після завершення аналізу застосунком формує рекомендації. Для цього використовується модуль `src/recommendations/engine.js`. Він аналізує розраховані метрики, статуси показників, benchmark-порівняння, *Z-score*, кореляції та динаміку періодів. На основі цих даних система створює рекомендаційні картки з аналітичним висновком, підтверджувальними сигналами та рекомендованою дією.

Окремим блоком реалізовано прогнозування. Після основного аналізу користувач може ввести новий денний бюджет і кількість днів прогнозу. Компонент прогнозу викликає відповідні функції з модуля `forecast.js`, після чого застосунком показує песимістичний, базовий та оптимістичний сценарії.

Візуалізація результатів реалізована через компоненти `charts.js` та `charts-block.js`. Вони відповідають за побудову графіків для порівняння фактичних значень із `benchmark`-показниками, аналізу динаміки за днями, порівняння кампаній або періодів, а також відображення сценаріїв прогнозу.

3.3 Реалізація введення, обробки та перевірки даних

3.3.1 Вибір параметрів аналізу

Перед введенням рекламних даних користувач обирає основні параметри аналізу. У застосунку цей етап реалізовано в початковому блоці розділу «Введення даних». Обрані параметри визначають, які саме дані буде обробляти система, які поля потрібно показати у формі, які метрики розраховувати та з якою базою порівняння зіставляти результати.

Користувач послідовно обирає задачу аналізу, тип рекламної кампанії, режим `benchmark`-порівняння, валюту та спосіб введення даних. Для задачі аналізу передбачено три режими: аналіз однієї кампанії, порівняння періодів рекламної кампанії та порівняння кількох кампаній. Такий поділ дозволяє використовувати застосунок як для швидкої оцінки однієї кампанії, так і для аналізу динаміки або порівняння декількох рекламних активностей між собою (рис. 3.2).

Після цього користувач обирає тип рекламної кампанії. У програмі підтримуються п'ять типів: Продажі, Лідогенерація, Трафік, Охоплення та Залучення. Вибір типу кампанії впливає на подальшу роботу застосунку, оскільки для кожного типу використовується власний набір вхідних полів, рекламних метрик, правил оцінювання, прогнозування та рекомендацій (рис. 3.2).

Введення даних
Оберіть задачу аналізу, тип реклами, бенчмарк, валюту та спосіб введення даних.

Оберіть задачу аналізу

- Аналіз однієї кампанії** (Selected): Розрахунок метрик і рекомендацій для однієї кампанії.
- Порівняння періодів рекламної кампанії: Порівняння кількох періодів однієї кампанії та зміна показників.
- Порівняння кампаній: Порівняння кількох кампаній одного типу між собою.

Оберіть тип реклами

- Продажі** (Selected): Покупки, дохід, прибуток і ROAS.
- Лідогенерація: Заявки, якісні ліди та CPL.
- Трафік: Кліки та перегляди посадкової сторінки.
- Охоплення: Охоплення та частота показів.
- Залучення: Взаємодія з контентом: реакції, коментарі, поширення.

Рисунок 3.2 — Вибір задачі та тип реклами

Далі користувач задає режим benchmark-порівняння. У застосунку доступні три варіанти: загальний benchmark, галузевий benchmark або режим без зовнішнього benchmark. Якщо обрано галузевий benchmark, інтерфейс додатково показує поле вибору галузі. У проекті передбачено такі галузі: краса, мода, електроніка, дім і сад, їжа та напої, автомобілі. Якщо зовнішній benchmark не використовується, аналіз може виконуватися на основі введених даних та внутрішньої бази порівняння, якщо вона доступна (рис. 3.3).

Оберіть benchmark для порівняння

- Загальний бенчмарк
- Галузевий бенчмарк** (Selected)
- Без зовнішнього бенчмарку

Оберіть галузь ⓘ

— оберіть галузь —

- оберіть галузь —
- Краса
- Мода
- Електроніка** (Selected)
- Дім і сад
- Їжа та напої
- Автомобілі

Рисунок 3.3 — Вибір benchmark

Також користувач обирає валюту (рис. 3.4). У програмі підтримуються UAH, USD і GBP. Валюта використовується для відображення грошових

показників, таких як Ad Spend, Revenue, Profit, CPA, CPC, CPM, CPL та інших вартісних метрик. При цьому розрахунки виконуються на основі введених числових значень, а вибрана валюта впливає на формат відображення результатів.

Останнім параметром є спосіб введення даних. У застосунку доступні два варіанти: ручне введення та CSV (рис 3.4). Якщо користувач обирає ручне введення, система показує форму з полями відповідно до типу кампанії. Якщо обрано CSV, відображається блок завантаження файлу, підказки щодо потрібних колонок і можливість завантажити шаблон.

Рисунок 3.4 — Вибір валюти та спосіб введення даних

З програмної точки зору вибір параметрів реалізовано через конфігураційні файли та компонент StepControls, який знаходиться у файлі `src/components/step-controls.js`. Основні доступні значення зберігаються у файлі `src/config/campaign-types.js`: типи кампаній, задачі аналізу, режими benchmark, способи введення даних та методи прогнозування. Завдяки цьому значення не дублюються в різних частинах коду, а використовуються як єдина конфігурація. Приклад визначення типів рекламних кампаній наведено у лістингу 3.1.

```
export const CAMPAIGN_TYPE = Object.freeze({
  SALES: 'sales',
  LEADS: 'leads',
  TRAFFIC: 'traffic',
  AWARENESS: 'awareness',
  ENGAGEMENT: 'engagement',
});
```

Лістинг 3.1 — Визначення типів рекламних кампаній

Компонент `StepControls` використовує ці значення для створення елементів вибору в інтерфейсі. Для кожного типу кампанії формується опція з внутрішнім значенням, назвою для користувача та короткою підказкою (лістинг 3.2)

```
const TYPE_OPTIONS = CAMPAIGN_TYPES.map((id) => ({
  value: id,
  label: dict.campaignType[id],
  hint: dict.campaignTypeHint[id],
}));
```

Лістинг 3.2 — Формування опцій вибору типу кампанії

Після вибору користувача спрацьовують відповідні обробники подій, наприклад `onTaskChange`, `onTypeChange`, `onBenchmarkModeChange`, `onCurrencyChange` або `onInputMethodChange`. Вони оновлюють стан застосунку, після чого інтерфейс перебудовується відповідно до нового вибору.

Для перевірки коректності вибраних параметрів використовується функція `stepsAreValid` (лістинг 3.3). Вона перевіряє, чи обрано основні параметри, а також чи вказана галузь у випадку використання галузевого `benchmark`.

```
export function stepsAreValid(state) {
  if (!state.task || !state.campaignType || !state.benchmarkMode) return
  false;
  if (state.benchmarkMode === BENCHMARK_MODE.INDUSTRY) {
    if (!state.industry || state.industry === '') return false;
  }
  return true;
}
```

Лістинг 3.3 — Перевірка вибраних параметрів аналізу

Якщо обов'язкові параметри не вибрані, застосунок не запускає аналіз. Наприклад, при виборі галузевого `benchmark` користувач обов'язково повинен обрати галузь. Це дозволяє уникнути ситуації, коли аналіз виконується без необхідної бази порівняння.

Таким чином, вибір параметрів аналізу реалізовано через поєднання конфігураційних файлів, стану застосунку та компонентів інтерфейсу.

3.3.2 Ручне введення даних

Ручне введення даних у застосунку використовується тоді, коли користувач хоче швидко проаналізувати одну кампанію, порівняти кілька кампаній або кілька періодів без завантаження CSV-файлу. Після вибору способу введення «Вручну» система показує форму з полями, які відповідають обраному типу рекламної кампанії.

Форма ручного введення реалізована у компоненті CampaignForm, який знаходиться у файлі `src/components/campaign-form.js`. Компонент не містить жорстко заданого набору полів для кожного типу кампанії. Потрібні поля беруться з конфігурації `FORM_FIELDS`, описаної у файлі `src/config/fields.js`. Завдяки цьому форма автоматично змінюється залежно від типу кампанії.

Основні поля ручного введення наведено в таблиці 3.3.

Таблиця 3.3 — Основні поля ручного введення

Тип кампанії	Основні поля
Продажі	Campaign Name, Start Date, End Date, Ad Spend, Revenue, Purchases, Impressions, Clicks, Reach, Frequency, Daily Budget
Лідогенерація	Campaign Name, Start Date, End Date, Ad Spend, Impressions, Clicks, Leads, Qualified Leads, Reach, Frequency, Daily Budget
Трафік	Campaign Name, Start Date, End Date, Ad Spend, Impressions, Clicks, Landing Page Views, Reach, Frequency, Daily Budget
Охоплення	Campaign Name, Start Date, End Date, Ad Spend, Reach, Impressions, Frequency, Target Audience Size, Daily Budget
Залучення	Campaign Name, Start Date, End Date, Ad Spend, Impressions, Reach, Engagements, Clicks, Comments, Shares, Saves, Frequency, Daily Budget

Не всі поля у формі є обов'язковими. Наприклад, Reach, Frequency, Daily Budget, Qualified Leads, Target Audience Size, Comments, Shares і Saves можуть бути додатковими залежно від типу кампанії. Якщо користувач не вводить додаткові дані, застосунок усе одно може виконати основний аналіз, але частина метрик, які залежать від цих полів, не буде розрахована.

Окрему логіку реалізовано для кампанії Продажі. Оскільки для оцінювання продажів важливо враховувати прибутковість, у формі передбачено вибір методу розрахунку прибутку: через повну собівартість або через середню валову маржу. Якщо користувач обирає перший варіант, у формі з'являється

поле Total Product Cost. Якщо обирається другий варіант, використовується поле Average Gross Margin. Це реалізовано через зміну активного поля прибутковості у компоненті CampaignForm.

У цьому фрагменті (лістинг 3.4) програма перевіряє, який метод розрахунку прибутковості вибрав користувач, і додає до форми відповідне поле. Таким чином, інтерфейс не показує два зайві поля одночасно, а відображає тільки той варіант, який потрібен для обраного способу розрахунку.

```
const activeProfitField =
  profitMethod === PROFIT_METHOD.MARGIN
    ? PROFITABILITY_FIELDS.margin
    : PROFITABILITY_FIELDS.cost;

node.appendChild(renderField(activeProfitField, values, currency, errors,
onChange, entryKey));
```

Лістинг 3.4 — Додавання поля прибутковості для кампанії Продажі

Для відображення полів використовуються компоненти NumberField і TextField з файлу src/components/form-controls.js. Числові поля мають відповідні обмеження: мінімальне значення, крок введення, вимогу цілого числа для кількісних показників, а також суфікси для валюти або відсотків. Наприклад, для грошових показників відображається символ обраної валюти, а для відсоткових показників знак %.

Під час введення значень стан форми оновлюється через обробник onFieldChange, який описаний у файлі src/app/orchestrator.js. Якщо користувач змінює значення поля, воно записується у відповідний об'єкт кампанії або періоду. Якщо раніше для цього поля була помилка, вона очищається після зміни значення. Це робить роботу з формою зручнішою, оскільки користувач одразу бачить, що некоректне поле було виправлене.

Якщо користувач обрав порівняння кампаній або періодів, у ручному режимі можна додавати кілька наборів даних. Для цього в інтерфейсі передбачені кнопки додавання кампанії або періоду (рис. 3.5, 3.6). Для аналізу

однієї кампанії потрібен один набір даних, а для порівняння кампаній або періодів щонайменше два.

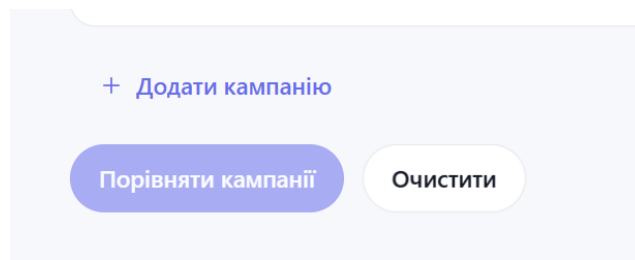


Рисунок 3.5 — Кнопка додавання кампанії

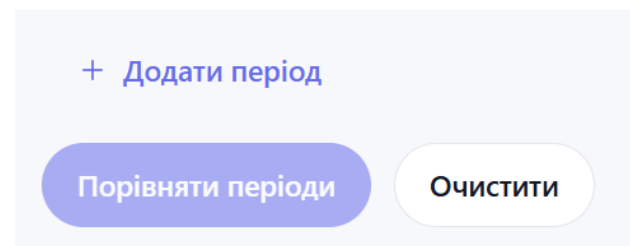


Рисунок 3.6 — Кнопка додавання періоду

Приклад форми ручного введення для кампанії Продажі наведено на рисунку 3.7.

Введення даних

Назва кампанії (Campaign Name)

Дата початку (Start Date)

Дата завершення (End Date)

Витрати на рекламу (Ad Spend)

Дохід (Revenue)

Покупки (Purchases)

Покази (Impressions)

Кліки (Clicks)

Охоплення (Reach) — необов'язкове

Частота (Frequency) — необов'язкове

Денний бюджет (Daily Budget) — необов'язкове

Метод розрахунку прибутковості: Через повну собівартість Через середню валову маржу

Загальна собівартість товарів (Total Product Cost)

Рисунок 3.7 — Форма ручного введення даних кампанії Продажі

Після заповнення форми та натискання кнопки аналізу введені вручну дані передаються до функції `normalizeManualEntries` у файлі `src/data/normalizer.js`. На цьому етапі виконується перевірка коректності

введення, приведення числових значень до потрібного формату, додавання інформації про метод прибутковості та підготовка структури даних для подальшого розрахунку метрик.

Отже, ручне введення в застосунку реалізовано як динамічна форма, що змінюється залежно від типу рекламної кампанії та вибраного режиму аналізу. Це дозволяє користувачу вводити тільки ті дані, які потрібні для конкретного типу кампанії, і не перевантажує інтерфейс зайвими полями.

3.3.3 Завантаження та обробка CSV-файлів

Окрім ручного введення, у веб-застосунку реалізовано можливість завантаження даних із CSV-файлу. У застосунку CSV-завантаження може використовуватися для аналізу всіх типів задач.

Блок завантаження CSV реалізовано через компонент CsvUpload, який знаходиться у файлі `src/components/csv-upload.js`. Користувач може обрати файл через стандартне вікно вибору файлу або перетягнути його в область завантаження (рис. 3.8). Для поля завантаження встановлено обмеження `accept=".csv,text/csv"`, тому інтерфейс орієнтований саме на CSV-файли.

Як підготувати CSV-файл

Для якісного аналізу рекомендується мати щоденну розбивку (daily breakdown).

ОБОВ'ЯЗКОВІ КОЛОНКИ

- Campaign Name
- Ad Spend
- Revenue
- Purchases
- Impressions
- Clicks

РЕКОМЕНДОВАНІ КОЛОНКИ

- Date
- Reach
- Frequency
- Daily Budget
- Average Gross Margin %
- Total Product Cost

ⓘ Якщо в CSV є колонка Date, система автоматично розпізнає щоденний розбивку і зможе побудувати Z-score, кореляційний аналіз та прогноз через лінійну регресію. Без дат доступне базове порівняння, але детальна аналітика буде обмежена.

⬇ Завантажити CSV-шаблон

Метод розрахунку прибутковості

Метод розрахунку прибутковості буде визначено автоматично після завантаження CSV.

Перетягніть CSV-файл сюди



Перетягніть CSV-файл сюди

або натисніть, щоб обрати файл. Перший рядок повинен містити назви колонок.

Усі грошові значення у CSV мають бути в одній вибраній валюті. Конвертація не виконується.

Проаналізувати кампанію

Очистити

Рисунок 3.8 — Блок завантаження CSV-файлу

Перед завантаженням файлу застосунок показує користувачу список обов'язкових і рекомендованих колонок. Цей блок реалізовано у компоненті CsvGuide у файлі `src/components/csv-guide.js`. Список обов'язкових колонок (таблиця 3.4) залежить від типу кампанії та береться з конфігурації `REQUIRED_CSV_COLUMNS` у файлі `src/config/fields.js`

Таблиця 3.4 — Обов'язкові CSV-колонки залежно від типу кампанії

Тип кампанії	Обов'язкові CSV-колонки
Продажі	campaignName, adSpend, revenue, purchases, impressions, clicks
Лідогенерація	campaignName, adSpend, impressions, clicks, leads
Трафік	campaignName, adSpend, impressions, clicks
Охоплення	campaignName, adSpend, reach, impressions
Залучення	campaignName, adSpend, impressions, engagements

Крім обов'язкових колонок, застосунок показує рекомендовані колонки. Наприклад, для Sales-кампаній рекомендованими є date, reach, frequency, dailyBudget, averageGrossMargin або totalProductCost. Наявність таких колонок не завжди є обов'язковою, але вони дозволяють розрахувати більше метрик, виконати аналіз динаміки за днями або побудувати точніший прогноз (рис. 3.8).

Для обробки CSV-файлів використовується бібліотека PapaParse. Її виклик реалізовано у модулі src/data/csv-parser.js. Під час парсингу вказано, що перший рядок файлу містить заголовки колонок, порожні рядки пропускаються, а значення не перетворюються автоматично, оскільки застосунок самостійно обробляє числові значення.

```

const config = {
  header: true,
  skipEmptyLines: 'greedy',
  dynamicTyping: false,
  transformHeader: (header) => header?.trim(),
  complete: (result) => {
    const parsed = process(result.data, result.meta ?? {}, campaignType);
    resolve(parsed);
  },
};

```

Лістинг 3.5 — Налаштування парсингу CSV-файлу

Після зчитування файлу застосунок нормалізує назви колонок. У проекті передбачено таблицю відповідностей `COLUMN_ALIASES`, яка дозволяє розпізнавати різні варіанти назв колонок англійською та українською мовами. Наприклад, поле `adSpend` може бути визначене за назвами `ad spend`, `spend`, `cost`, `amount spent`, витрати, витрати на рекламу тощо. Це зроблено для того, щоб застосунок міг працювати з файлами, експортованими з різних рекламних платформ або підготовленими користувачем вручну.

Після розпізнавання заголовків система перевіряє, чи присутні в CSV-файлі обов'язкові колонки для обраного типу кампанії (лістинг 3.6). Якщо хоча б одна обов'язкова колонка відсутня, файл не приймається, а користувач отримує повідомлення про помилку.

```

const requiredIds = REQUIRED_CSV_COLUMNS[campaignType] || [];
const presentIds = new Set(Object.values(headerMap));

for (const required of requiredIds) {
  if (!presentIds.has(required)) {
    const label = headerLabel(required);
    throw new Error(t('validation.csvMissingColumn')(label));
  }
}

```

Лістинг 3.6 — Перевірка наявності обов'язкових CSV-колонок

Після первинної перевірки CSV-дані передаються до функції `normalizeCsv` у файлі `src/data/normalizer.js`. На цьому етапі рядки файлу перетворюються у внутрішню структуру застосунку. Якщо файл містить кілька кампаній, дані можуть групуватися за назвою кампанії. Якщо виконується порівняння періодів, дані можуть групуватися за назвою періоду. Для цього використовується допоміжна функція `bucketByName`.

Якщо у CSV-файлі є колонка `date`, застосунок визначає файл як такий, що містить денну деталізацію. У цьому випадку рядки зберігаються не тільки в агрегованому вигляді, а й окремо як денні дані. Надалі вони використовуються для побудови графіків динаміки, `Z-score`, кореляційного аналізу та регресійного прогнозування.

Під час агрегації застосунок підсумовує кількісні показники, такі як `adSpend`, `revenue`, `purchases`, `impressions`, `clicks`, `leads`, `landingPageViews`, `engagements`, `comments`, `shares` і `saves`. Деякі значення, наприклад `campaignName`, `periodName`, `startDate`, `endDate`, `targetAudienceSize`, `dailyBudget` або `frequency`, беруться з доступних рядків як окремі параметри. Якщо є `impressions` і `reach`, частота показів може бути визначена як відношення показів до охоплення.

Для кампаній типу Продажі при CSV-завантаженні також враховується метод розрахунку прибутковості. Якщо у файлі є колонка `totalProductCost`, застосунок використовує метод через повну собівартість. Якщо є колонка `averageGrossMargin`, використовується метод через середню валову маржу. Якщо потрібних даних для прибутковості немає, система показує помилку і не запускає аналіз.

У режимі порівняння кампаній або періодів CSV-завантаження може працювати двома способами: через один спільний файл або через кілька окремих файлів (рис. 3.9). Це реалізовано в компоненті `data-input.js`, де для таких задач користувач може перемикати між режимами `singleFile` і `multipleFiles`. У першому

випадку система намагається розділити дані за назвами кампаній або періодів, у другому кожен файл відповідає окремій кампанії або періоду.

Після успішної обробки файлу система показує статус завантаження. Якщо файл містить помилки, наприклад відсутні обов'язкові колонки, некоректні дати або не логічні значення, застосунок не передає ці дані до аналізу. Це дозволяє уникнути неправильних розрахунків і забезпечує коректну роботу наступних модулів.

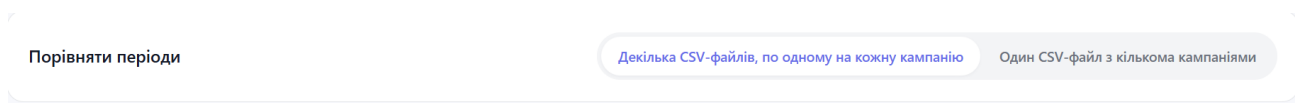


Рисунок 3.9 — Вибір способу завантаження CSV-файла

3.3.4 Перевірка коректності даних

Перед запуском аналізу застосунок перевіряє коректність введених або завантажених даних. Це потрібно для того, щоб система не виконувала розрахунки на основі порожніх, некоректних або логічно неможливих значень. Перевірка даних реалізована у файлі `src/lib/validation.js`, а для ручного введення додатково викликається через функцію `normalizeManualEntries` у файлі `src/data/normalizer.js`.

Перевірка виконується у два етапи. На першому етапі система перевіряє окремі поля: чи заповнені обов'язкові значення, чи є введене значення числом, чи відповідає воно допустимому діапазону, а також чи має воно бути цілим числом. Наприклад, кількість показів, кліків, покупок, лідів або взаємодій повинна бути цілим числом, а витрати на рекламу мають бути додатним числом.

```
export function validateNumberField(fieldDef, rawValue) {
```

```

const isEmpty = rawValue === null || rawValue === undefined || rawValue
=== '';
if (isEmpty) {
  return fieldDef.required ? t('validation.required')(label) : null;
}

const num = typeof rawValue === 'number' ? rawValue :
toNumber(rawValue);

if (!Number.isFinite(num)) {
  return t('validation.number')(label);
}

if (fieldDef.integer && !Number.isInteger(num)) {
  return t('validation.integer')(label);
}

return null;
}

```

Лістинг 3.7 — Перевірка числового поля

Тут показано основну логіку перевірки числового поля. Якщо поле є обов’язковим і не заповнене, система повертає повідомлення про помилку. Якщо значення не можна перетворити на число або воно має бути цілим, але введено дробове значення, аналіз також не запускається.

Окремо перевіряються поля дат Start Date і End Date які не є обов’язковими для всіх сценаріїв, але якщо користувач заповнює одне з них, система вимагає заповнити й інше. Також перевіряється, щоб дата завершення не була раніше дати початку. Це потрібно для коректного визначення тривалості кампанії або періоду.

Для кампаній типу Продажі додатково перевіряються дані, потрібні для розрахунку прибутковості. Якщо користувач обрав метод через повну собівартість, поле Total Product Cost стає обов’язковим. Якщо обрано метод через середню валову маржу, перевіряється поле Gross Margin, причому його значення має бути більшим за 0 і не перевищувати 100%.

На другому етапі виконується перехресна перевірка показників. Вона потрібна для виявлення логічно неможливих співвідношень між метриками.

Наприклад, кількість кліків не може бути більшою за кількість показів, охоплення не може перевищувати кількість показів, а кількість покупок у кампанії Продажі не повинна бути більшою за кількість кліків.

```

    if (has('clicks') && has('impressions') && data.clicks >
data.impressions) {
        errors.push(t('validation.clicksImpressions'));
    }

    if (has('reach') && has('impressions') && data.reach > data.impressions)
{
        errors.push(t('validation.reachImpressions'));
    }

    if (campaignType === CAMPAIGN_TYPE.SALES) {
        if (has('purchases') && has('clicks') && data.clicks > 0 &&
data.purchases > data.clicks) {
            errors.push(t('validation.purchasesClicks'));
        }
    }
}

```

Лістинг 3.8 — Перехресна перевірка пов'язаних показників

Подібні перевірки реалізовані й для інших типів кампаній. Для Лідогенерації система перевіряє, щоб кількість лідів не перевищувала кількість кліків, а кількість кваліфікованих лідів не була більшою за загальну кількість лідів. Для кампанії типу Трафік перевіряється, щоб кількість переглядів цільової сторінки не перевищувала кількість кліків. Для Охоплення розмір цільової аудиторії не може бути меншим за охоплення. Для Залучення кількість коментарів, поширень і збережень не повинна перевищувати загальну кількість взаємодій.

Для ручного введення перевірка запускається після натискання кнопки аналізу. Якщо знайдено помилки, вони зберігаються у стані застосунку як `fieldErrors` або `globalErrors`. Помилки окремих полів відображаються безпосередньо біля відповідного поля форми, а загальні помилки показуються окремим повідомленням у блоці введення даних (рисунок 3.9, 3.10). Якщо

користувач змінює значення поля, попередня помилка для цього поля очищається.

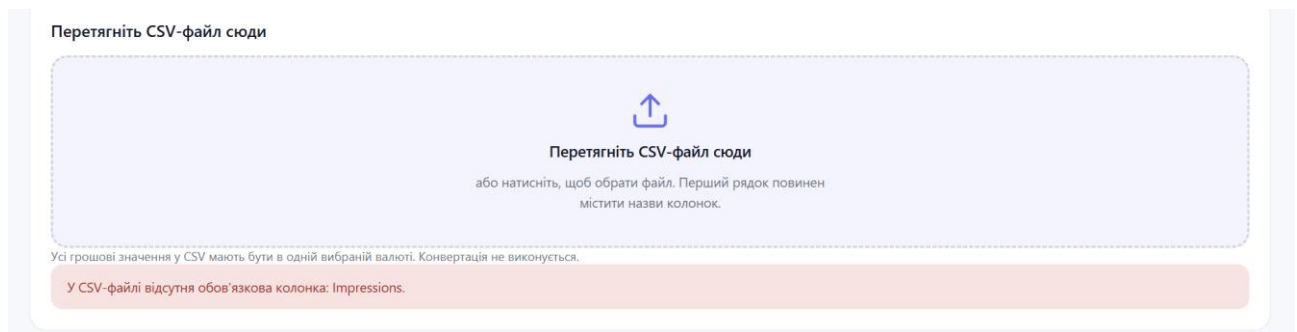


Рисунок 3.9 — Приклад статусу помилки під час завантаження CSV-файлу

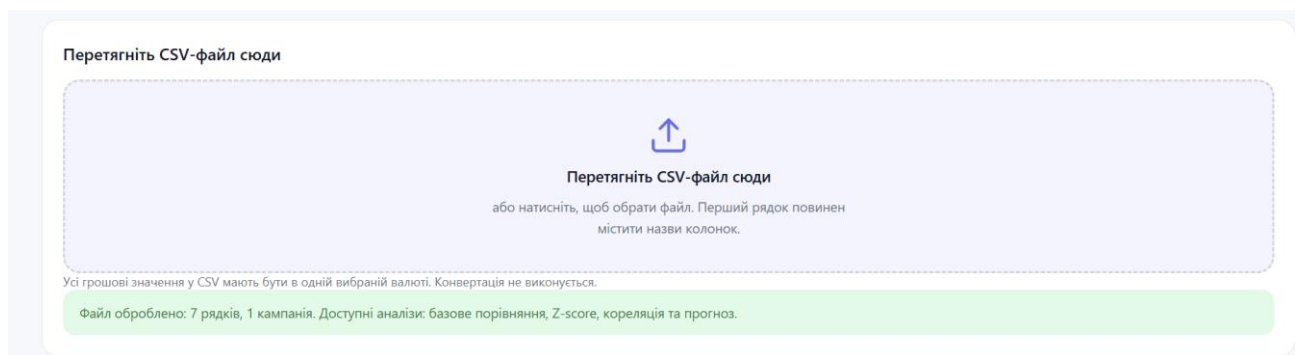


Рисунок 3.10 — Приклад статусу успіху під час завантаження CSV-файлу

Для CSV-файлів перевірка починається ще на етапі обробки заголовків. Застосунок перевіряє наявність обов'язкових колонок для обраного типу кампанії. Якщо потрібної колонки немає, файл не приймається, а користувач бачить повідомлення про помилку. Після цього система перевіряє значення у рядках, дати та логічні зв'язки між показниками так само, як і при ручному введенні.

У процесі запуску аналізу також перевіряється, чи достатньо наборів даних для вибраної задачі. Для аналізу однієї кампанії достатньо одного набору даних, а для порівняння кампаній або періодів потрібно щонайменше два набори. Якщо користувач обрав режим порівняння, але надав лише один набір даних, застосунок не запускає аналіз і показує відповідне повідомлення.

```

if (state.task !== ANALYSIS_TASK.SINGLE && entities.length < 2) {
  store.setState({
    globalErrors: [
      state.task === ANALYSIS_TASK.PERIODS
        ? 'Для порівняння періодів потрібно мінімум 2 набори даних.'
        : 'Для порівняння кампаній потрібно мінімум 2 набори даних.',
    ],
    analysis: null,
    recommendations: null,
  });
  return false;
}

```

Лістинг 3.9 — Перевірка кількості наборів даних для порівняння

3.3.5 Нормалізація даних перед аналізом

Після перевірки коректності дані не передаються одразу до розрахунку метрик. Спочатку вони проходять етап нормалізації, тобто приводяться до єдиного внутрішнього формату, з яким можуть працювати аналітичні модулі застосунку. Нормалізація реалізована у файлі `src/data/normalizer.js`.

Для ручного введення використовується функція `normalizeManualEntries`. Вона отримує тип кампанії, метод розрахунку прибутковості, список введених форм та тип задачі аналізу. Після успішної перевірки значення числових полів приводяться до числового формату, додається інформація про метод прибутковості, визначається наявність денного бюджету та формується об'єкт кампанії або періоду. Поле `profitMethod` зберігає спосіб розрахунку прибутковості для кампаній Продажі, а `dailyBudgetProvided` показує, чи був заданий денний бюджет. Після цього викликається функція `applyDateDerivedFields`, яка розраховує додаткові поля, пов'язані з датами та бюджетом.

```

const coerced = coerceNumericFields(fields, entry.data);
coerced.profitMethod = profitMethod;
coerced.dailyBudgetProvided = isFiniteNumber(coerced.dailyBudget) &&
coerced.dailyBudget > 0;
applyDateDerivedFields(coerced);

```

Лістинг 3.10 — Нормалізація даних ручного введення

Для CSV-даних нормалізація є складнішою, оскільки файл може містити один або декілька рядків, денну деталізацію, кілька кампаній або кілька періодів. Для цього використовується функція `normalizeCsv`. Якщо користувач аналізує кілька кампаній, рядки можуть групуватися за назвою кампанії. Якщо виконується порівняння періодів, рядки можуть групуватися за назвою періоду.

```

if (task === ANALYSIS_TASK.CAMPAIGNS && parsed.campaignNames.length > 1)
{
    entities = bucketByName(parsed.rows, 'campaignName', task);
} else if (task === ANALYSIS_TASK.PERIODS && parsed.periodNames.length >
1) {
    entities = bucketByName(parsed.rows, 'periodName', task);
}

```

Лістинг 3.11 — Групування CSV-рядків за кампанією або періодом

Після групування рядки CSV-файлу агрегуються. Кількісні показники, такі як `adSpend`, `revenue`, `purchases`, `impressions`, `clicks`, `leads`, `landingPageViews`, `engagements`, `comments`, `shares` і `saves`, підсумовуються. Це дозволяє отримати загальні значення для кампанії або періоду на основі кількох рядків.

```

for (const field of SUM_FIELDS) {
  let sum = 0;
  let any = false;

  for (const row of rows) {
    const v = row[field];
    if (isFiniteNumber(v)) {
      sum += v;
      any = true;
    }
  }

  totals[field] = any ? sum : null;
}

```

Лістинг 3.12 — Агрегація кількісних показників CSV-файлу

Частина параметрів не підсумовується, а береться з доступних рядків як окреме значення. До таких полів належать `campaignName`, `periodName`, `startDate`, `endDate`, `targetAudienceSize`, `dailyBudget`, `frequency`, `totalProductCost` та `averageGrossMargin`. Якщо в CSV-файлі є `impressions` і `reach`, але частота показів не задана явно, застосунок може визначити `frequency` як відношення показів до охоплення.

Якщо CSV-файл містить колонку `date`, застосунок зберігає денну деталізацію окремо у полі `daily`. Це потрібно для побудови графіків динаміки, `Z-score`, кореляційного аналізу та регресійного прогнозування. Тобто агреговані значення використовуються для загального аналізу кампанії, а денні рядки залишаються доступними для аналізу змін у часі.

Окремо нормалізуються дані, пов'язані з датами. Якщо є денні рядки, застосунок автоматично визначає дату початку, дату завершення та тривалість за кількістю унікальних дат. Якщо денних рядків немає, але користувач ввів `Start Date` і `End Date`, тривалість визначається на основі цього періоду.

```

if (dailyRows?.length) {
  const dates = Array.from(new Set(dailyRows.map((r) =>
r.date).filter(Boolean))).sort();
  if (dates.length) {
    input.startDate ??= dates[0];
    input.endDate ??= dates[dates.length - 1];
    input.durationDays = dates.length;
  }
}

```

Лістинг 3.13 — Визначення тривалості кампанії за датами

Після визначення тривалості застосунок може розрахувати додаткові бюджетні показники. Якщо задано фактичні витрати та тривалість, система визначає `actualDailySpend`. Якщо користувач також задав `Daily Budget`, застосунок розраховує планові витрати, різницю між фактичними і плановими витратами, відсоток відхилення та співвідношення фактичного бюджету до планового.

Для кампаній типу Продажі нормалізація також враховує спосіб розрахунку прибутковості. Якщо використовується середня валова маржа, вона може переноситися в денні рядки. Якщо використовується повна собівартість, а CSV містить денну деталізацію, собівартість може розподілятися між денними рядками пропорційно до доходу. Це потрібно, щоб денні метрики для Sales-кампаній могли коректно розраховувати прибутковість.

Після завершення нормалізації застосунок формує масив `entities`. Кожен елемент цього масиву відповідає окремій кампанії або періоду і містить назву, ключ, агреговані вхідні дані та, за наявності, денну деталізацію. Саме цей масив передається до модуля аналізу для подальшого розрахунку рекламних метрик.

3.4 Реалізація розрахунку рекламних метрик та аналітичних показників

Після введення, перевірки та нормалізації даних застосунк переходить до розрахунку рекламних метрик і аналітичних показників. Цей етап реалізовано в окремих модулях папки `src/calc`, де зосереджено логіку обчислення метрик, benchmark-порівняння, статусів, Z-score, кореляційного аналізу, Campaign Score та узагальнення результатів.

3.4.1 Розрахунок рекламних метрик за типами кампаній

Основний розрахунок рекламних метрик реалізовано у файлі `src/calc/metrics.js`. У застосунку не використовується одна універсальна формула для всіх кампаній, оскільки різні типи реклами мають різні цілі. Тому для кожного типу кампанії створено окрему функцію розрахунку. Вибір потрібної функції виконується через об'єкт `CALCULATORS`, де кожному типу кампанії відповідає свій розрахунковий модуль.

```
const CALCULATORS = {
  [CAMPAIGN_TYPE.SALES]: salesMetrics,
  [CAMPAIGN_TYPE.LEADS]: leadsMetrics,
  [CAMPAIGN_TYPE.TRAFFIC]: trafficCampaignMetrics,
  [CAMPAIGN_TYPE.AWARENESS]: awarenessMetrics,
  [CAMPAIGN_TYPE.ENGAGEMENT]: engagementMetrics,
};
export function computeMetrics(campaignType, input) {
  const calculator = CALCULATORS[campaignType];
  if (!calculator) {
    throw new Error(`Unknown campaign type: ${campaignType}`);
  }
  return calculator(input);
}
```

Лістинг 3.14 — Агрегація кількісних показників CSV-файлу

Такий підхід дозволяє розділити логіку розрахунків за типами кампаній і не перевантажувати одну функцію великою кількістю умов.

3.4.2 Захист від некоректних обчислень

Оскільки більшість рекламних метрик розраховується через ділення, у застосунку передбачено захист від некоректних обчислень. Для цього використовується файл `src/calc/safe.js`, у якому реалізовано допоміжні функції `safeDiv`, `safeMul`, `safeSub`, `safeAdd`.

Перед виконанням арифметичних операцій система перевіряє, чи є значення коректними числами. Для цього використовується функція `isFiniteNumber`.

```
export function isFiniteNumber(value) {
  return typeof value === 'number' && Number.isFinite(value);
}
```

Лістинг 3.15 — Перевірка числового значення

Функція `safeDiv` використовується для захищеного ділення. Вона перевіряє чисельник і знаменник, а також не дозволяє виконувати ділення на нуль. Якщо обчислення неможливе, функція повертає `null`.

```
export function safeDiv(numerator, denominator) {
  if (!isFiniteNumber(numerator) || !isFiniteNumber(denominator)) return
  null;
  if (denominator === 0) return null;
  return numerator / denominator;
}
```

Лістинг 3.16 — функція `safeDiv`

Функція `safeMul` використовується для множення одного або кількох значень. Якщо хоча б одне зі значень не є коректним числом, результат також повертається як `null`.

```
export function safeMul(...values) {
  let acc = 1;
  for (const v of values) {
    if (!isFiniteNumber(v)) return null;
    acc *= v;
  }
  return acc;
}
```

Лістинг 3.17 — функція `safeMul`

Функція `safeSub` використовується для віднімання. Вона повертає результат тільки в тому випадку, якщо обидва операнди є коректними числами.

```
export function safeSub(a, b) {
  if (!isFiniteNumber(a) || !isFiniteNumber(b)) return null;
  return a - b;
}
```

Лістинг 3.18 — функція `safeSub`

Функція `safeAdd` використовується для додавання кількох значень. Якщо хоча б одне передане значення не є числом, результатом буде `null`.

```
export function safeAdd(...values) {
  let acc = 0;
  for (const v of values) {
    if (!isFiniteNumber(v)) return null;
    acc += v;
  }
  return acc;
}
```

Лістинг 3.19 — функція `safeAdd`

Окремо у файлі `safe.js` реалізовано функцію `safeSum`, яка підсумовує тільки коректні числові значення з масиву. Якщо в масиві немає жодного коректного числа, функція повертає `null`.

```
export function safeSum(values) {
  let total = 0;
  let any = false;
  for (const v of values) {
    if (isFiniteNumber(v)) {
      total += v;
      any = true;
    }
  }
  return any ? total : null;
}
```

Лістинг 3.20 — Підсумовування доступних числових значень

Також у цьому модулі є функція `num`, яка використовується для перетворення текстових значень у числа. Вона прибирає символи валют, відсотків, пробіли та обробляє різні формати десяткового роздільника. Це особливо важливо при роботі з CSV-файлами, де числа можуть бути записані у різному форматі.

```
export function num(value) {
  if (typeof value === 'number') return Number.isFinite(value) ? value :
null;

  if (value == null || value === '') return null;
  if (typeof value !== 'string') return null;

  let str = value.trim();
  if (!str) return null;

  str = str.replace(/[%@\$€£¥₽]/g, '');
  str = str.replace(/[\s ]/g, '');

  const n = Number(str);
  return Number.isFinite(n) ? n : null;
}
```

Лістинг 3.21 — Перетворення текстового значення у число

Як видно з рисунка 3.11, після запуску аналізу застосунок відображає основні метрики кампанії у вигляді зведених карток. Для кожного показника показується його значення, коротке пояснення та статус, який допомагає швидко оцінити результат кампанії.

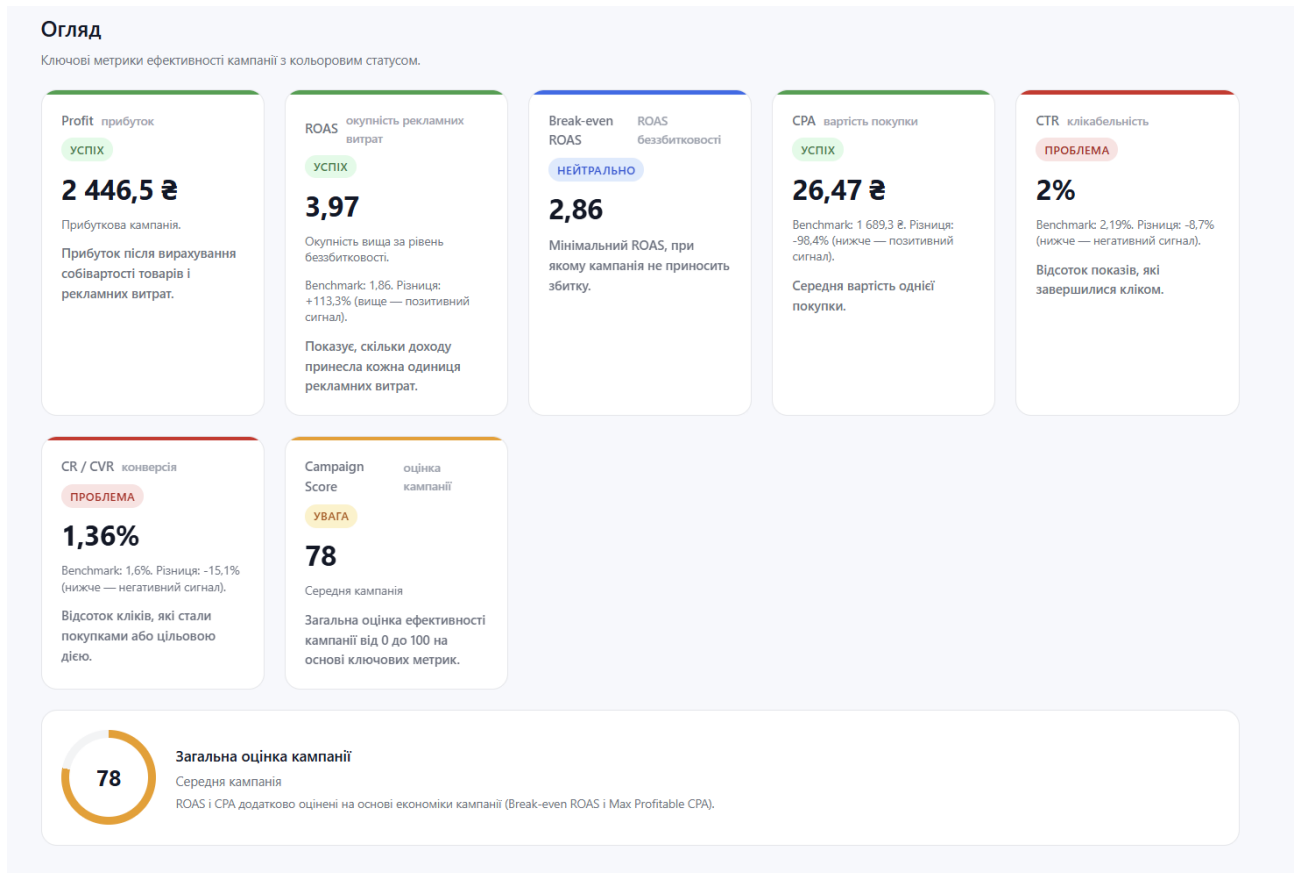


Рисунок 3.11 — Зведені метрики кампанії після запуску аналізу

3.4.3 Реалізація benchmark-порівняння та статусів показників

Після розрахунку метрик застосунок порівнює їх із зовнішніми benchmark-показниками або внутрішніми середніми значеннями.

Benchmark-показники зберігаються у файлі `src/config/benchmarks.js`. У цьому файлі визначено значення для різних типів кампаній, а також джерела benchmark-даних, зокрема Triple Whale 2026 та WordStream / LocaliQ 2025. Для вартісних показників у benchmark-конфігурації також зберігається валюта, оскільки частина значень задана в USD.

Пошук потрібного benchmark-значення реалізовано у файлі `src/calc/benchmark-lookup.js` через функцію `resolveBenchmark`. Вона отримує контекст аналізу: тип кампанії, режим `benchmark`, галузь і валюту. Якщо користувач обрав режим без зовнішнього `benchmark`, функція одразу повертає `null`.

```
export function resolveBenchmark(ctx, metricId) {
  if (ctx.benchmarkMode === BENCHMARK_MODE.NONE) return null;
  const metricDef = meta(metricId);
  const benchKey = metricDef.benchKey;
  if (!benchKey) return null;

  const typeBucket = BENCHMARKS[ctx.campaignType];
  if (!typeBucket) return null;
}
```

Лістинг 3.22 — Пошук benchmark-значення для метрики

Тут початку перевіряється, чи взагалі потрібно використовувати зовнішній `benchmark`. Після цього система визначає ключ `benchmark`-показника

для конкретної метрики та знаходить набір benchmark-значень для обраного типу кампанії.

У застосунку передбачено два режими зовнішнього порівняння: загальний benchmark і галузевий benchmark. Якщо користувач обирає загальний benchmark, система бере значення з блоку overall. Якщо обрано галузевий benchmark, система спочатку шукає значення для конкретної галузі. Якщо точного галузевого значення немає, застосунок може використати загальний benchmark як fallback.

```

if (ctx.benchmarkMode === BENCHMARK_MODE.OVERALL) {
  const item = tryGet('overall');
  if (!item) return null;
  return buildResult(item, null, false, ctx.currency);
}

if (ctx.benchmarkMode === BENCHMARK_MODE.INDUSTRY && ctx.industry) {
  const item = tryGet(ctx.industry);
  if (item) {
    return buildResult(item, ctx.industry, false, ctx.currency);
  }

  const overall = tryGet('overall');
  if (!overall) return null;
  return buildResult(overall, null, true, ctx.currency);
}

```

Лістинг 3.23 — Вибір загального або галузевого benchmark

Окремо реалізовано конвертацію benchmark-значень для вартісних метрик. Якщо benchmark задано в USD, а користувач обрав UAH або GBP, значення перераховується за курсами, заданими у конфігурації EXCHANGE_RATES.

```
if (item.currency === 'USD' && targetCurrency === 'UAH') {
  return { value: item.value * EXCHANGE_RATES.USD_UAH, currency: 'UAH' };
}

if (item.currency === 'USD' && targetCurrency === 'GBP') {
  return {
    value: (item.value * EXCHANGE_RATES.USD_UAH) / EXCHANGE_RATES.GBP_UAH,
    currency: 'GBP',
  };
}
```

Лістинг 3.24 — Конвертація benchmark-значення у валюту користувача

Після отримання benchmark-значення застосунок визначає статус метрики. Ця логіка реалізована у файлі `src/calc/status.js`. У програмі використовуються такі статуси: `success`, `warning`, `danger` та `info`. Статус `success` означає, що показник є прийнятним або кращим за базу порівняння. Статус `warning` використовується для прикордонних значень, `danger` для проблемних показників, а `info` для нейтральних або недоступних оцінок.

Для метрик, де вище значення є кращим, використовується функція `higherBetter`. Вона порівнює фактичне значення з `benchmark` або внутрішнім середнім значенням. Якщо база порівняння відсутня, система може враховувати `Z-score`.

```
function higherBetter({ value, bench, internal, z }) {
  if (!isFiniteNumber(value)) return NEUTRAL;
  const targets = collectTargets(bench, internal);

  if (targets.length === 0) {
    if (isFiniteNumber(z)) {
      if (z >= 1) return SUCCESS;
      if (z <= -2) return DANGER;
      if (z <= -1) return WARNING;
      return SUCCESS;
    }
    return NEUTRAL;
  }

  const worst = strict(value, targets, '>=');
  if (worst === 'above') return SUCCESS;
  if (worst === 'close') return WARNING;
  if (isFiniteNumber(z) && z <= -2) return DANGER;
  return DANGER;
}
```

Лістинг 3.25 — Оцінювання метрик, де вище значення є кращим

Для метрик, де нижче значення є кращим, використовується функція `lowerBetter`. Її логіка подібна, але напрям порівняння протилежний: значення нижче `benchmark` або внутрішнього середнього вважається кращим.

```

function lowerBetter({ value, bench, internal, z }) {
  if (!isFiniteNumber(value)) return NEUTRAL;
  const targets = collectTargets(bench, internal);

  if (targets.length === 0) {
    if (isFiniteNumber(z)) {
      if (z <= -1) return SUCCESS;
      if (z >= 2) return DANGER;
      if (z >= 1) return WARNING;
      return SUCCESS;
    }
    return NEUTRAL;
  }

  const worst = strict(value, targets, '<=');
  if (worst === 'above') return SUCCESS;
  if (worst === 'close') return WARNING;
  if (isFiniteNumber(z) && z >= 2) return DANGER;
  return DANGER;
}

```

Лістинг 3.26 — Оцінювання метрик, де нижче значення є кращим

Щоб не позначати незначні відхилення як проблему, у застосунку використовується допуск `CLOSE_TOLERANCE = 0.05`, тобто 5%. Якщо фактичне значення трохи відрізняється від `benchmark`, система може встановити статус `warning`, а не одразу `danger`.

Для кампаній Продажі частина статусів має окрему економічну логіку. Наприклад, `ROAS` оцінюється відносно `Break-even ROAS`, а `CPA` відносно `Max Profitable CPA`. Це важливо, тому що для продажів не завжди достатньо порівняти показник із загальним `benchmark`: кампанія може мати `CPA` вищий за середній рівень, але все ще залишатися прибутковою, якщо маржа та середній чек це дозволяють.

```
function statusRoas({ value, ctx }) {
  if (!isFiniteNumber(value)) return NEUTRAL;
  if (isFiniteNumber(ctx?.profit) && ctx.profit < 0) return DANGER;
  if (!isFiniteNumber(ctx?.breakEvenRoas)) return NEUTRAL;

  const diff = (value - ctx.breakEvenRoas) / Math.abs(ctx.breakEvenRoas);
  if (diff >= CLOSE_TOLERANCE) return SUCCESS;
  if (diff > -CLOSE_TOLERANCE) return WARNING;
  return DANGER;
}
```

Лістинг 3.27 — Оцінювання ROAS відносно рівня беззбитковості

Відповідність метрик конкретним функціям оцінювання задається через об'єкт HANDLERS. У ньому для кожного типу кампанії визначено, які метрики потрібно оцінювати та яким способом.

```
const HANDLERS = {
  [CAMPAIGN_TYPE.SALES]: {
    profit: statusProfit,
    roas: statusRoas,
    cpa: statusCpaAgainstMax,
    ctr: higherBetter,
    cr: higherBetter,
    campaignScore: statusCampaignScore,
  },
  [CAMPAIGN_TYPE.TRAFFIC]: {
    ctr: higherBetter,
    cpc: lowerBetter,
    cpm: lowerBetter,
    lpvRate: higherBetter,
    costPerLpv: lowerBetter,
    campaignScore: statusCampaignScore,
  },
};
```

Лістинг 3.28 — Оцінювання ROAS відносно рівня беззбитковості

Відповідність метрик конкретним функціям оцінювання задається через об'єкт HANDLERS. У ньому для кожного типу кампанії визначено, які метрики потрібно оцінювати та яким способом

```
const HANDLERS = {
  [CAMPAIGN_TYPE.SALES]: {
    profit: statusProfit,
    roas: statusRoas,
    cpa: statusCpaAgainstMax,
    ctr: higherBetter,
    cr: higherBetter,
    campaignScore: statusCampaignScore,
  },
  [CAMPAIGN_TYPE.TRAFFIC]: {
    ctr: higherBetter,
    cpc: lowerBetter,
    cpm: lowerBetter,
    lpvRate: higherBetter,
    costPerLpv: lowerBetter,
    campaignScore: statusCampaignScore,
  },
};
```

Лістинг 3.29 — Прив'язка метрик до функцій оцінювання

Цей об'єкт містить правила для всіх типів реклами. На рисунку 3.12 показано порівняння з benchmark в інтерфейсі:



Рисунок 3.12 — Реалізація benchmark-порівняння та статусів показників

3.4.4 Реалізація статистичного аналізу

Статистичний аналіз у застосунку використовується для виявлення відхилень у рекламних показниках, аналізу зв'язків між метриками та порівняння результатів між кампаніями або періодами. Основні статистичні функції реалізовано у файлі `src/calc/statistics.js`, а їх використання в загальному аналізі відбувається через модуль `src/calc/analysis.js`.

Статистичний аналіз у застосунку використовується для виявлення відхилень у рекламних показниках, аналізу зв'язків між метриками та порівняння результатів між кампаніями або періодами. Основні статистичні функції реалізовано у файлі `src/calc/statistics.js`, а їх використання в загальному аналізі відбувається через модуль `src/calc/analysis.js`.

У файлі `statistics.js` реалізовано розрахунок середнього значення, стандартного відхилення, Z-score, кореляції Пірсона, лінійної регресії та відсоткової зміни. Перед розрахунками дані очищуються: функції працюють тільки з коректними числовими значеннями, а `null`, `undefined` або некоректні значення ігноруються.

```
export function mean(values) {
  const arr = cleanSingle(values);
  if (arr.length === 0) return null;
  let sum = 0;
  for (const v of arr) sum += v;
  return sum / arr.length;
}
```

Лістинг 3.30 — Розрахунок середнього значення

Стандартне відхилення використовується для подальшого розрахунку Z-score. Якщо немає жодного коректного значення, функція повертає `null`.

```

export function stdDev(values) {
  const arr = cleanSingle(values);
  if (arr.length === 0) return null;
  const m = mean(arr);
  let acc = 0;
  for (const v of arr) {
    const d = v - m;
    acc += d * d;
  }
  return Math.sqrt(acc / arr.length);
}

```

Лістинг 3.31 — Розрахунок стандартного відхилення

Z-score у застосунку використовується для визначення того, наскільки значення метрики відхиляється від середнього рівня. Для коректного розрахунку потрібно мінімум два значення. Якщо значень недостатньо або стандартне відхилення дорівнює нулю, функція повертає null.

```

export function zScoresOf(values) {
  const arr = cleanSingle(values);
  if (arr.length < 2) return values.map(() => null);
  const m = mean(arr);
  const sd = stdDev(arr);
  if (!isFiniteNumber(sd) || sd === 0) return values.map(() => null);
  return values.map((v) => (isFiniteNumber(v) ? (v - m) / sd : null));
}

```

Лістинг 3.32 — Розрахунок Z-score для набору значень

У модулі analysis.js Z-score розраховується для кампаній або періодів, якщо їх більше одного. Для кожної числової метрики формується набір значень, після чого застосовується функція zScoresOf.

```
function computeZScores(campaigns) {
  const out = {};
  if (campaigns.length < 2) return out;
  const fields = collectFields(campaigns);

  for (const field of fields) {
    const values = campaigns.map((c) => c.merged[field]);
    const zs = zScoresOf(values);
    zs.forEach((z, i) => {
      if (!out[i]) out[i] = {};
      out[i][field] = z;
    });
  }

  return out;
}
```

Лістинг 3.33 — Розрахунок Z-score між кампаніями або періодами

Якщо у даних є денна деталізація, застосунок додатково розраховує денні Z-score. Це дозволяє знаходити дні, у які певна метрика суттєво відхилялася від звичайного рівня кампанії. Наприклад, можна виявити день з різким зростанням CPA, CPM або падінням CTR. Відображення блоку Z-score аналізу в інтерфейсі (рис. 3.13).

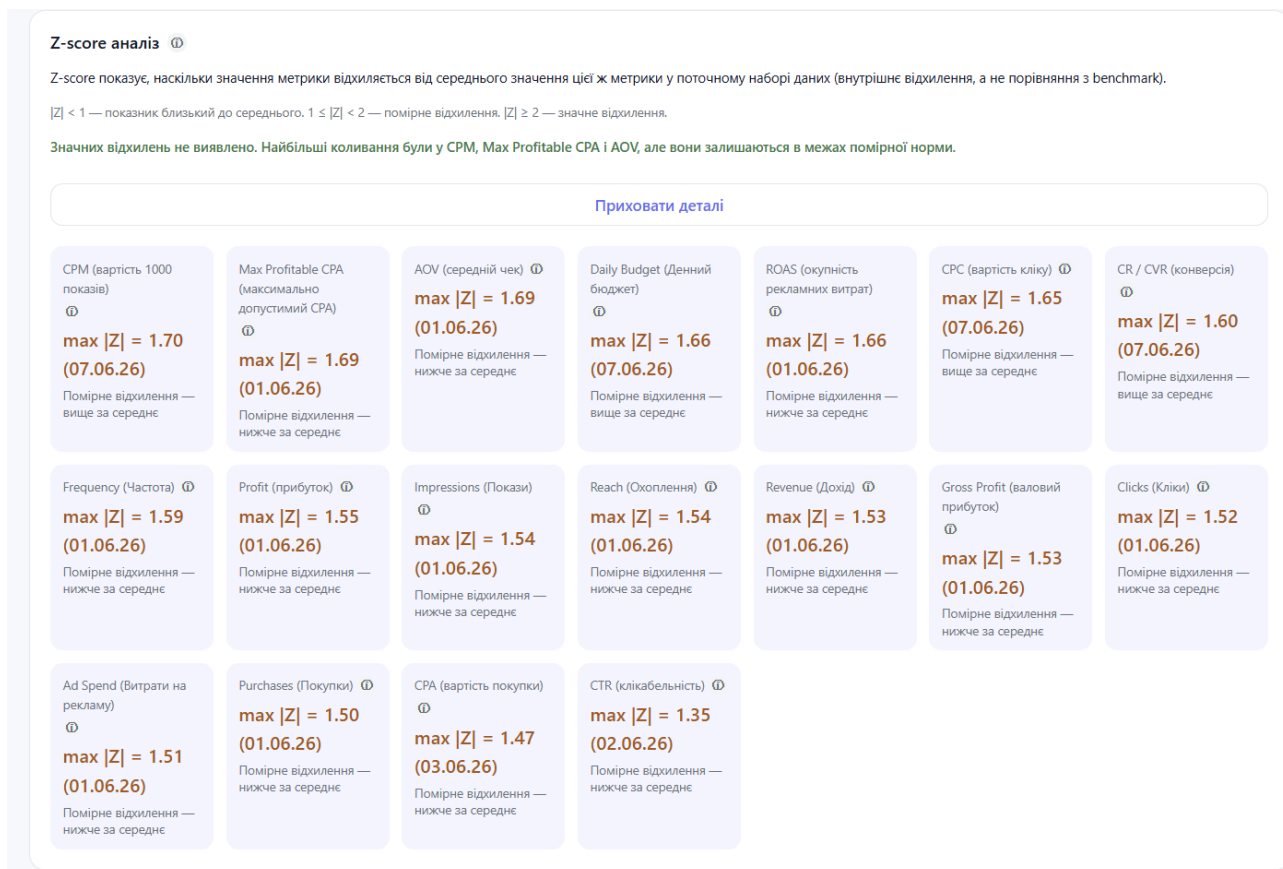


Рисунок 3.13 — Відображення Z-score аналізу в інтерфейсі

Кореляційний аналіз реалізовано у файлі `src/calc/correlations.js`. Для кожного типу кампанії задано власні пари метрик, між якими є сенс перевіряти статистичний зв'язок.

```

export const CORRELATION_PAIRS = {
  [CAMPAIGN_TYPE.SALES]: [
    ['frequency', 'ctr'],
    ['cpc', 'cpa'],
    ['cr', 'cpa'],
    ['roas', 'profit'],
    ['adSpend', 'profit'],
    ['adSpend', 'purchases'],
  ],
  [CAMPAIGN_TYPE.LEADS]: [
    ['frequency', 'ctr'],
    ['ctr', 'cpl'],
    ['leadCr', 'cpl'],
    ['qualifiedLeadRate', 'costPerQualifiedLead'],
    ['adSpend', 'leads'],
  ],
  [CAMPAIGN_TYPE.TRAFFIC]: [
    ['ctr', 'cpc'],
    ['cpm', 'cpc'],
    ['lpvRate', 'costPerLpv'],
    ['frequency', 'ctr'],
    ['adSpend', 'landingPageViews'],
  ],
  [CAMPAIGN_TYPE.AWARENESS]: [
    ['adSpend', 'reach'],
    ['frequency', 'reach'],
    ['cpm', 'costPer1000Reach'],
    ['frequency', 'costPer1000Reach'],
  ],
  [CAMPAIGN_TYPE.ENGAGEMENT]: [
    ['frequency', 'engagementRate'],
    ['cpm', 'costPerEngagement'],
    ['engagementRate', 'costPerEngagement'],
    ['adSpend', 'engagements'],
    ['ctr', 'engagementRate'],
  ],
};

```

Лістинг 3.34 — Пари метрик для кореляційного аналізу за типами кампаній

Сам розрахунок кореляції Пірсона виконується у файлі `statistics.js`. Для розрахунку використовується тільки набір коректних пар числових значень. Якщо таких пар менше трьох, кореляція не розраховується.

```
export function pearson(xs, ys) {
  const pairs = cleanPairs(xs, ys);
  if (pairs.length < 3) return null;

  const mx = pairs.reduce((s, [x]) => s + x, 0) / pairs.length;
  const my = pairs.reduce((s, [, y]) => s + y, 0) / pairs.length;

  let cov = 0;
  let sx = 0;
  let sy = 0;

  for (const [x, y] of pairs) {
    const dx = x - mx;
    const dy = y - my;
    cov += dx * dy;
    sx += dx * dx;
    sy += dy * dy;
  }

  const denom = Math.sqrt(sx * sy);
  if (denom === 0) return null;
  return cov / denom;
}
```

Лістинг 3.35 — Розрахунок коефіцієнта кореляції Пірсона

Після розрахунку коефіцієнт кореляції додатково інтерпретується за силою зв'язку.

```

export function correlationStrength(r) {
  if (!isFiniteNumber(r)) return null;
  const abs = Math.abs(r);
  if (abs < 0.3) return 'weak';
  if (abs < 0.7) return 'moderate';
  return 'strong';
}

```

Лістинг 3.36 — Визначення сили кореляційного зв'язку

Кореляція може розраховуватися по-різному залежно від доступних даних. Якщо є денна деталізація, система використовує денні значення. Якщо аналізуються кілька кампаній без денних даних, кореляція може розраховуватися за агрегованими показниками різних кампаній або періодів. Для кореляції потрібні щонайменше три коректні точки даних, тому при меншій кількості спостережень застосунок не показує кореляційний результат. Відображення кореляційного аналізу в інтерфейсі (рис. 3.14).

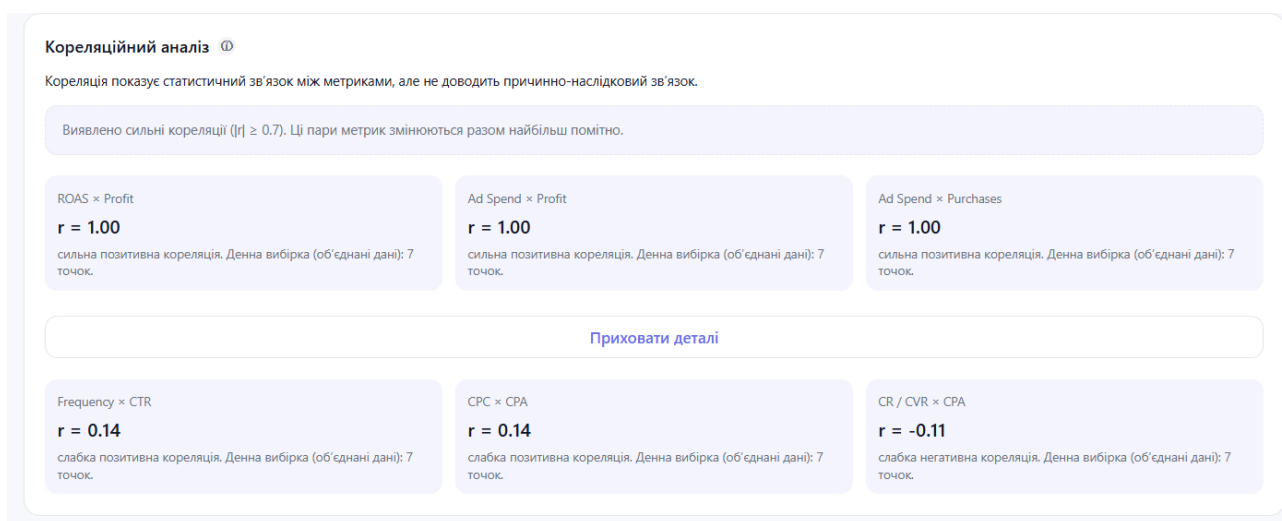


Рисунок 3.14 — Відображення кореляційного аналізу в інтерфейсі

3.4.5 Реалізація Campaign Score

Для узагальненої оцінки ефективності рекламної кампанії у застосунку реалізовано показник Campaign Score. Він відображає загальну оцінку кампанії в діапазоні від 0 до 100 і дозволяє користувачу швидко зрозуміти, наскільки добре кампанія працює за ключовими метриками. Програмна реалізація Campaign Score знаходиться у файлі `src/calc/campaign-score.js`, а ваги метрик задані у файлі `src/config/score-weights.js`.

```
export const SCORE_WEIGHTS = Object.freeze({
  [CAMPAIGN_TYPE.SALES]: {
    full: W([
      ['profit', 0.4],
      ['roas', 0.25],
      ['cpa', 0.2],
      ['ctr', 0.075],
      ['cr', 0.075],
    ]),
  },
  [CAMPAIGN_TYPE.ENGAGEMENT]: {
    full: W([
      ['engagementRate', 0.4],
      ['costPerEngagement', 0.3],
      ['ctr', 0.15],
      ['frequency', 0.15],
    ]),
  },
});
```

Лістинг 3.37 — Ваги Campaign Score для різних типів кампаній

У лістингу 3.37 наведено скорочений фрагмент. У проєкті також задано ваги для Lead Generation, Traffic та Awareness, а для деяких типів кампаній передбачено додаткові схеми.

Вибір потрібної схеми виконується функцією `pickPreset`. Вона перевіряє, які дані доступні для кампанії, і повертає відповідний набір ваг.

```
function pickPreset(campaignType, ctx) {
  const presets = SCORE_WEIGHTS[campaignType];
  if (!presets) return null;

  if (campaignType === CAMPAIGN_TYPE.SALES) {
    return ctx.hasProfit ? presets.full : presets.noProfit ??
presets.full;
  }

  if (campaignType === CAMPAIGN_TYPE.LEADS) {
    return ctx.hasQualifiedLeads ? presets.full : presets.noQualified ??
presets.full;
  }

  if (campaignType === CAMPAIGN_TYPE.AWARENESS) {
    return ctx.hasAudience ? presets.full : presets.noAudience ??
presets.full;
  }

  return presets.full;
}
```

Лістинг 3.38 — Вибір схеми ваг для Campaign Score

Для кожної метрики розраховується часткова оцінка. Якщо для показника кращим є більше значення, фактичне значення ділиться на цільове. Якщо для показника кращим є менше значення, використовується обернене співвідношення. Результат обмежується значенням 100, щоб одна дуже сильна метрика не створювала необмежено високий внесок.

```

function ratioScore(metricId, actual, target) {
  if (!isFiniteNumber(actual)) return null;
  if (!isFiniteNumber(target) || target === 0) return null;

  const m = meta(metricId);

  if (m.direction === 'higher_better') {
    return Math.min((actual / target) * 100, 100);
  }

  if (m.direction === 'lower_better') {
    return Math.min((target / actual) * 100, 100);
  }

  return null;
}

```

Лістинг 3.39 — Розрахунок часткової оцінки метрики

Для метрики Profit використовується окрема функція profitScore, яка оцінює прибуток відносно рекламних витрат. Якщо прибуток значно перевищує витрати, оцінка наближається до 100. Якщо кампанія збиткова, оцінка знижується.

```

export function profitScore(profit, adSpend) {
  if (!Number.isFinite(profit) || !Number.isFinite(adSpend) || adSpend <=
0) {
    return null;
  }

  const ratio = profit / adSpend;

  if (ratio >= 2) return 100;
  if (ratio <= -1) return 0;

  if (ratio < 0) return 50 + ratio * 50;
  return 50 + (ratio / 2) * 50;
}

```

Лістинг 3.39 — Оцінювання прибутковості кампанії

Окремо обробляється метрика Frequency.

```
export function frequencyScore(frequency) {
  if (!Number.isFinite(frequency) || frequency < 0) return null;
  if (frequency < 1) return 50;
  if (frequency <= 3) return 100;
  if (frequency <= 6) return 100 - ((frequency - 3) / 3) * 40;
  if (frequency <= 10) return 60 - ((frequency - 6) / 4) * 30;
  return 20;
}
```

Лістинг 3.40 — Оцінювання Frequency

Після розрахунку часткових оцінок застосунків об'єднує їх у загальну оцінку. Якщо деякі метрики неможливо розрахувати, вони не беруть участь у фінальному обчисленні, а ваги доступних метрик перенормовуються. Це дозволяє отримати оцінку навіть тоді, коли користувач не ввів частину додаткових даних.

```
const totalWeight = raw.reduce((s, r) => s + r.weight, 0);
let total = 0;

for (const r of raw) {
  const renormalized = r.weight / totalWeight;
  total += r.score * renormalized;
  r.normWeight = renormalized;
}
```

Лістинг 3.41 — Оцінювання Frequency

У застосунку також передбачено корекцію ситуацій, коли одна дуже сильна метрика може приховати слабкість інших показників. Для цього система визначає слабкі та сильні часткові оцінки.

```
function applyBalanceCorrection(weightedScore, contributions, weak,
strong) {
  if (weak.length === 0 || strong.length === 0) return weightedScore;

  const blend = weak.length === 1 ? 0.25 : weak.length === 2 ? 0.40 : 0.50;
  const weakAvg = weak.reduce((s, c) => s + c.score, 0) / weak.length;

  return weightedScore * (1 - blend) + weakAvg * blend;
}
```

Лістинг 3.42 — Корекція змішаних сигналів

Після всіх розрахунків значення Campaign Score обмежується діапазоном від 0 до 100. Разом із фінальним значенням система також зберігає внесок окремих метрик, пропущені показники, кількість слабких і сильних сторін. Ці дані надалі використовуються для відображення оцінки в інтерфейсі та формування пояснень. Відображення Campaign Score в інтерфейсі (рис. 3.15).

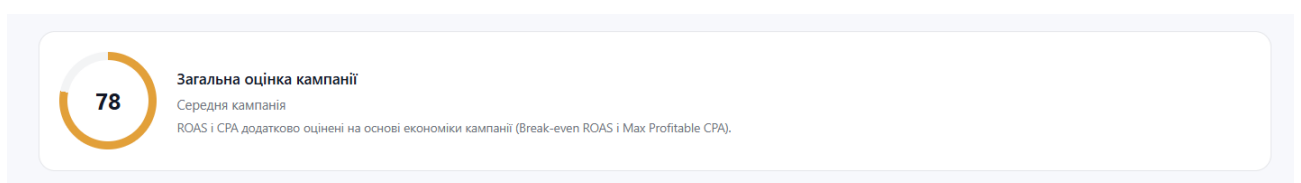


Рисунок 3.15 — Відображення Campaign Score в інтерфейсі

3.4.4 Узагальнення результатів аналізу

Після розрахунку окремих метрик, статусів, статистичних показників і Campaign Score застосунок повинен об'єднати всі результати в єдину структуру. Це потрібно для того, щоб різні блоки інтерфейсу могли працювати з однаковими даними: зведені картки, детальний аналіз, рекомендації, графіки та прогноз. Узагальнення результатів реалізовано у файлі `src/calc/analysis.js`.

Головною функцією цього модуля є `analyze`. Вона отримує нормалізовані дані, тип задачі, тип кампанії, режим `benchmark`, валюту, галузь і метод розрахунку прибутковості. На основі цих параметрів функція формує підсумковий об'єкт аналізу.

Першим етапом у функції `analyze` є створення масиву `campaigns`. Для кожної кампанії або періоду застосунок бере вхідні дані, розраховує метрики через `computeMetrics`, а потім об'єднує початкові дані й розраховані показники в об'єкт `merged`.

```
const campaigns = args.entities.map((entity, index) => {

  const inputs = entity.inputs;
  const metrics = computeMetrics(args.campaignType, inputs);
  const daily = entity.daily ? buildDaily(args.campaignType, entity.daily)
: null;

  return {
    key: entity.key,
    name: entity.name,
    index,
    inputs,
    metrics,
    merged: mergeInputsAndMetrics(inputs, metrics),
    daily,
  };
});
```

Лістинг 3.43 — Формування об'єкта кампанії під час аналізу

Якщо для кампанії доступна денна деталізація, вона також обробляється окремо. Для цього використовується функція `buildDaily`, яка агрегує рядки за датами, розраховує метрики для кожного дня та формує денні об'єкти з полем `merged`.

```
function buildDaily(campaignType, rows) {
  const aggregated = aggregateByDate(rows);
  return aggregated.map((row) => {
    const metrics = computeMetrics(campaignType, row);
    return { ...row, metrics, merged: mergeInputsAndMetrics(row, metrics)
  });
};
}
```

Лістинг 3.44 — Побудова денних даних для аналізу

Після формування кампаній система розраховує внутрішні середні значення та стандартні відхилення. Це виконується тоді, коли аналізується більше однієї кампанії або більше одного періоду. Такі значення використовуються як внутрішня база порівняння, якщо зовнішній benchmark не застосовується або якщо потрібно оцінити відхилення між об'єктами аналізу.

```
const internalAverages = {};
const internalStdDevs = {};

if (campaigns.length > 1) {
  const fields = collectFields(campaigns);
  for (const field of fields) {
    const values = campaigns.map((c) => c.merged[field]);
    internalAverages[field] = mean(values);
    internalStdDevs[field] = stdDev(values);
  }
}
```

Лістинг 3.45 — Розрахунок внутрішніх середніх значень

Далі застосунок розраховує Z-score, відсоткові зміни між періодами, порівняння періодів або кампаній, benchmark-порівняння, статуси показників і Campaign Score. Для кожної метрики статус визначається через функцію resolveStatus, а комплексна оцінка кампанії через computeCampaignScore.

```

    for (const metricId of statusMetricIds) {
        campaign.statuses[metricId] = resolveStatus(args.campaignType,
metricId, {
            value: campaign.merged[metricId],
            bench: comparisonBenchResolver(metricId),
            internal: comparisonInternalResolver(metricId),
            z: zScores[campaign.index]?.[metricId] ?? null,
            ctx: statusCtx,
        });
    }

    campaign.score = applyPeriodScoreAdjustment(computeCampaignScore({
        campaignType: args.campaignType,
        metrics: campaign.merged,
        inputs: campaign.inputs,
        resolveBench: comparisonBenchResolver,
        resolveInternal: comparisonInternalResolver,
        benchmarkMode: args.benchmarkMode,
    }), periodComparison?.byIndex?.[campaign.index]);

```

Лістинг 3.46 — Формування статусів і Campaign Score

Після цього виконується кореляційний аналіз і формується детальний Z-score аналіз. Якщо є денні дані, система може аналізувати відхилення за днями. Якщо денних даних немає, але є кілька кампаній або періодів, застосунок використовує агреговані значення.

У кінці функція `analyze` повертає єдиний об'єкт з усіма результатами. Цей об'єкт містить інформацію про задачу аналізу, тип кампанії, валюту, режим `benchmark`, масив кампаній, внутрішні середні значення, Z-score, кореляції, порівняння періодів або кампаній, а також функції для отримання `benchmark` і внутрішніх значень.

```
return {  
  task: args.task,  
  campaignType: args.campaignType,  
  benchmarkMode: args.benchmarkMode,  
  industry: args.industry,  
  currency: args.currency,  
  profitMethod: args.profitMethod,  
  campaigns,  
  internalAverages,  
  internalStdDevs,  
  zScores,  
  correlationAnalysis,  
  dailyZScores,  
  zScoreAnalysis,  
  deltas,  
  periodComparison,  
  campaignComparison,  
  benchResolver,  
  internalResolver,  
};
```

Лістинг 3.47 — Повернення підсумкового об'єкта аналізу

Після завершення функції `analyze` результат передається до `orchestrator.js`. Там він зберігається у стані застосунку як `analysis`. На основі цього самого об'єкта одразу формується блок рекомендацій через функцію `generateRecommendations`, а також визначається, чи доступний регресійний прогноз.

```

recommendations = generateRecommendations(analysis) ?? {};

store.setState({
  processing: false,
  analysis,
  recommendations,
  activeEntityIndex: 0,
  forecast: {
    ...currentForecast,
    results: null,
    regressionAvailable,
  },
});

```

Лістинг 3.48 — Збереження результатів аналізу у стані застосунку

Приклад формування Рекомендацій в інтерфейсі (рис. 3.16)

Рекомендації

Діагностичні висновки з урахуванням доступних benchmark-метрик, Z-score, кореляції і динаміки.

Кампанія є прибутковою

АНАЛІТИЧНИЙ ВИСНОВОК
Критичних проблем із прибутковістю не виявлено.

ПІДТВЕРДЖУВАЛЬНІ СИГНАЛИ

- Profit є позитивним.
- ROAS вищий за рівень беззбитковості.
- CPA нижчий за максимально допустиму вартість покупки.

РЕКОМЕНДОВАНІ ДІЇ
Продовжуйте моніторинг кампанії. Рішення про масштабування бюджету прийміть після перевірки стабільності ROAS, CPA, CTR, Frequency і динаміки прибутку.

Кампанія потребує комплексної оптимізації

ОСНОВНА ПРОБЛЕМА
Кампанія має проблеми на двох етапах: залучення кліків і конверсія після переходу.

ПІДТВЕРДЖУВАЛЬНІ СИГНАЛИ

- CTR нижчий за benchmark або внутрішню базу порівняння.
- CR / CVR нижчий за benchmark або внутрішню базу порівняння.

РЕКОМЕНДОВАНІ ДІЇ
Не масштабуйте кампанію на цьому етапі. Спочатку протестуйте нові креативи та рекламні повідомлення, потім перевірте релевантність посадкової сторінки, ціну, умови доставки та процес оформлення.

Кампанія прибуткова попри низький CTR

АНАЛІТИЧНИЙ ВИСНОВОК
Кампанія є прибутковою, але CTR нижчий за вибраний benchmark. Це означає, що економічно кампанія працює, але може мати потенціал для покращення залучення.

ПІДТВЕРДЖУВАЛЬНІ СИГНАЛИ

- Profit є позитивним.
- ROAS вищий за рівень беззбитковості.
- CPA нижчий за максимально допустимий CPA.
- CTR нижчий за benchmark або внутрішню базу порівняння.

РЕКОМЕНДОВАНІ ДІЇ
Не зупиняйте кампанію лише через низький CTR. Обережно тестуйте нові креативи та відстежуйте, чи покращення CTR також покращує CPA, ROAS і прибуток.

Рисунок 3.13 — Приклад блоку Рекомендації

3.5 Реалізація модуля прогнозування

3.5.1 Вхідні параметри прогнозу

Модуль прогнозування у вебзастосунку доступний після виконання основного аналізу рекламної кампанії. Відображення блоку прогнозування реалізовано у компоненті `ForecastBlock`, який знаходиться у файлі `src/components/forecast.js`.

У формі прогнозу користувач вводить два основні параметри: новий денний бюджет і кількість днів прогнозу.

```
const form = el('div', { class: 'field-grid' }, [
  NumberField({
    field: budgetField,
    value: forecast.newDailyBudget ?? '',
    currency: analysis.currency,
    fieldKey: 'forecast',
    onChange: (v) => onChange?.({ newDailyBudget: v }),
  }),
  NumberField({
    field: { id: 'forecastDays', required: true, min: 1, max: 365,
integer: true, inclusive: true },
    value: forecast.forecastDays ?? '',
    fieldKey: 'forecast',
    onChange: (v) => onChange?.({ forecastDays: v }),
  }),
]);
```

Лістинг 3.49 — Поля введення параметрів прогнозу

Для нового денного бюджету використовується числове поле з відображенням валюти, яку користувач обрав на етапі введення даних. Це дозволяє зберігати єдиний формат грошових показників у всьому застосунку. Значення кількості днів також вводиться як числове поле, але для нього встановлено вимогу цілого числа.

Крім параметрів бюджету та тривалості, користувач може обрати метод прогнозування через рекламну воронку або на основі лінійної регресії.

Вибір методу реалізовано через масив `methodOptions`. За замовчуванням у ньому завжди є прогноз через воронку. Якщо регресія доступна, до списку додається другий варіант прогнозу на основі лінійної регресії.

```
const methodOptions = [
  {
    value: FORECAST_METHOD.FUNNEL,
    label: t('forecastMethod.funnel'),
    tooltip: t('forecastMethodHint.funnel'),
  },
];

if (effectiveRegressionAvailable) {
  methodOptions.push({
    value: FORECAST_METHOD.REGRESSION,
    label: t('forecastMethod.reggression'),
    tooltip: t('forecastMethodHint.reggression'),
  });
}
```

Лістинг 3.50 — Формування доступних методів прогнозування

Доступність регресійного прогнозу визначається після основного аналізу в модулі `orchestrator.js`. Система перевіряє, чи є у кампанії денна історія, чи аналізуються кілька періодів або кілька кампаній. Якщо таких даних достатньо, прапорець `regressionAvailable` встановлюється як `true`.

```
const hasDailyHistory = Boolean(forecastBase?.daily &&
forecastBase.daily.length >= 2);

const hasPeriodsHistory =
  analysis.task === ANALYSIS_TASK.PERIODS && analysis.campaigns.length >=
2;

const hasCampaignComparisonHistory =
  analysis.task === ANALYSIS_TASK.CAMPAIGNS && analysis.campaigns.length
>= 2;

const regressionAvailable =
  hasDailyHistory || hasPeriodsHistory || hasCampaignComparisonHistory;
```

Лістинг 3.51 — Визначення доступності регресійного прогнозу

Додатково в компоненті ForecastBlock уточнюється, чи доступна регресія саме для активної кампанії.

Перед запуском прогнозу введені значення нормалізуються у функції runForecast. Новий денний бюджет і кількість днів перетворюються на числові значення. Також до вхідних параметрів передається targetAudienceSize, якщо він є у вихідних даних кампанії, оскільки цей показник може використовуватися для прогнозу кампаній Охоплення.

```
const normalizedInputs = {
  newDailyBudget: Number(forecastInputs.newDailyBudget),
  forecastDays: Number(forecastInputs.forecastDays),
  targetAudienceSize: campaign.inputs.targetAudienceSize,
};
```

Лістинг 3.52 — Нормалізація вхідних параметрів прогнозу

Після цього функція runForecast визначає, який метод прогнозування потрібно використати. Якщо користувач обрав регресію і вона доступна, система намагається сформувати набір історичних спостережень. Якщо регресійний набір даних недоступний, застосунок автоматично використовує прогноз через рекламну воронку.

```

if (
  forecastInputs.method === FORECAST_METHOD.REGRESSION &&
  forecastInputs.regressionAvailable
) {
  const regressionDataset = selectRegressionDataset(analysis, campaign);
  if (!regressionDataset) {
    return funnelForecastScenarios({
      campaignType: analysis.campaignType,
      metrics: campaign.metrics,
      inputs: campaign.inputs,
      forecastInputs: normalizedInputs,
    });
  }

  return regressionForecastScenarios({
    campaignType: analysis.campaignType,
    observations: regressionDataset.observations,
    current: regressionDataset.current ?? campaign,
    forecastInputs: normalizedInputs,
    isDaily: regressionDataset.isDaily,
  });
}

```

Лістинг 3.53 — Вибір методу прогнозування

Блок форми для введення параметрів прогнозування (рис. 3.14).

Прогноз
Простий прогноз через воронку, сценарії та лінійна регресія, якщо доступні історичні дані.

Денний бюджет (Daily Budget)

Кількість днів прогнозу (Forecast Days)

Метод прогнозу

Прогноз через воронку Прогноз на основі лінійної регресії

Рисунок 3.14 — Форма введення параметрів прогнозу

3.5.2 Прогноз через рекламну воронку

Прогноз через рекламну воронку реалізовано у файлі `src/calc/forecast.js`. Цей метод використовується як базовий варіант прогнозування і доступний після запуску аналізу кампанії. Його логіка полягає в тому, що система бере поточні коефіцієнти кампанії та застосовує їх до нового бюджету, який задає користувач.

```
function preflightFunnel(input) {
  const { newDailyBudget, forecastDays } = input;
  if (!isFiniteNumber(newDailyBudget) || newDailyBudget <= 0) return
null;
  if (!isFiniteNumber(forecastDays) || forecastDays <= 0) return null;
  return newDailyBudget * forecastDays;
}
```

Лістинг 3.54 — Розрахунок прогнозних витрат

Після цього система вибирає функцію прогнозу відповідно до типу кампанії. Для цього у файлі `forecast.js` використовується об'єкт `FUNNELS`, де кожному типу кампанії відповідає окрема функція.

```
const FUNNELS = {
  [CAMPAIGN_TYPE.SALES]: forecastSales,
  [CAMPAIGN_TYPE.LEADS]: forecastLeads,
  [CAMPAIGN_TYPE.TRAFFIC]: forecastTraffic,
  [CAMPAIGN_TYPE.AWARENESS]: forecastAwareness,
  [CAMPAIGN_TYPE.ENGAGEMENT]: forecastEngagement,
};
```

Лістинг 3.55 — Вибір функції воронкового прогнозу

Базові коефіцієнти для прогнозу беруться з уже розрахованих метрик кампанії.

```
function baseCoeffs(campaignType, metrics, inputs) {
  const base = {
    cpm: metrics.cpm,
    ctr: metrics.ctr,
    cr: metrics.cr,
    leadCr: metrics.leadCr,
    lpvRate: metrics.lpvRate,
    engagementRate: metrics.engagementRate,
    aov: metrics.aov,
    grossMargin: metrics.grossMargin,
    qualifiedLeadRate: metrics.qualifiedLeadRate,
    frequency: metrics.frequency,
  };

  if (campaignType === CAMPAIGN_TYPE.SALES && inputs.profitMethod ===
PROFIT_METHOD.MARGIN) {
    base.grossMargin = inputs.averageGrossMargin;
  }

  return base;
}
```

Лістинг 3.56 — Формування базових коефіцієнтів прогнозу

```

function forecastSales(input, coeffs) {
  const newAdSpend = preflightFunnel(input);
  if (newAdSpend == null) return null;

  const { cpm, ctr, cr, aov, grossMargin } = coeffs;
  if (!isFiniteNumber(cpm) || cpm <= 0) return null;

  const predictedImpressions = safeMul(safeDiv(newAdSpend, cpm), 1000);
  const predictedClicks = safeMul(predictedImpressions, pct(ctr));
  const predictedPurchases = safeMul(predictedClicks, pct(cr));
  const predictedRevenue = safeMul(predictedPurchases, aov);
  const predictedGrossProfit = safeMul(predictedRevenue,
    pct(grossMargin));

  const predictedProfit =
    isFiniteNumber(predictedGrossProfit) && isFiniteNumber(newAdSpend)
      ? predictedGrossProfit - newAdSpend
      : null;

  const predictedRoas = safeDiv(predictedRevenue, newAdSpend);
  const predictedCpa = safeDiv(newAdSpend, predictedPurchases);

  return {
    predictedSpend: newAdSpend,
    predictedImpressions,
    predictedClicks,
    predictedPurchases,
    predictedRevenue,
    predictedGrossProfit,
    predictedProfit,
    predictedRoas,
    predictedCpa,
  };
}

```

Лістинг 3.57 — Воронковий прогноз для кампанії Пролажі.

```

function forecastLeads(input, coeffs) {
  const newAdSpend = preflightFunnel(input);
  if (newAdSpend == null) return null;

  const { cpm, ctr, leadCr, qualifiedLeadRate } = coeffs;
  if (!isFiniteNumber(cpm) || cpm <= 0) return null;

  const predictedImpressions = safeMul(safeDiv(newAdSpend, cpm), 1000);
  const predictedClicks = safeMul(predictedImpressions, pct(ctr));
  const predictedLeads = safeMul(predictedClicks, pct(leadCr));
  const predictedCpl = safeDiv(newAdSpend, predictedLeads);

  let predictedQualifiedLeads = null;
  let predictedCostPerQualifiedLead = null;

  if (isFiniteNumber(qualifiedLeadRate)) {
    predictedQualifiedLeads = safeMul(predictedLeads,
    pct(qualifiedLeadRate));
    predictedCostPerQualifiedLead = safeDiv(newAdSpend,
    predictedQualifiedLeads);
  }

  return {
    predictedSpend: newAdSpend,
    predictedImpressions,
    predictedClicks,
    predictedLeads,
    predictedCpl,
    predictedLeadCr: isFiniteNumber(leadCr) ? leadCr : null,
    predictedQualifiedLeads,
    predictedCostPerQualifiedLead,
  };
}

```

Лістинг 3.58 — Воронковий прогноз для Лідогенерації.

```
function forecastTraffic(input, coeffs) {
  const newAdSpend = preflightFunnel(input);
  if (newAdSpend == null) return null;

  const { cpm, ctr, lpvRate } = coeffs;
  if (!isFiniteNumber(cpm) || cpm <= 0) return null;

  const predictedImpressions = safeMul(safeDiv(newAdSpend, cpm), 1000);
  const predictedClicks = safeMul(predictedImpressions, pct(ctr));
  const predictedLandingPageViews = safeMul(predictedClicks,
pct(lpvRate));
  const predictedCpc = safeDiv(newAdSpend, predictedClicks);
  const predictedCostPerLpv = safeDiv(newAdSpend,
predictedLandingPageViews);

  return {
    predictedSpend: newAdSpend,
    predictedImpressions,
    predictedClicks,
    predictedLandingPageViews,
    predictedCpc,
    predictedLpvRate: isFiniteNumber(lpvRate) ? lpvRate : null,
    predictedCostPerLpv,
  };
}
```

Лістинг 3.59 — Воронковий прогноз для Трафік.

```

function forecastAwareness(input, coeffs) {
  const newAdSpend = preflightFunnel(input);
  if (newAdSpend == null) return null;

  const { cpm, frequency } = coeffs;
  if (!isFiniteNumber(cpm) || cpm <= 0) return null;
  if (!isFiniteNumber(frequency) || frequency <= 0) return null;

  const predictedImpressions = safeMul(safeDiv(newAdSpend, cpm), 1000);
  const predictedReach = safeDiv(predictedImpressions, frequency);
  const predictedCostPer1000Reach = safeMul(safeDiv(newAdSpend,
predictedReach), 1000);

  let predictedReachRate = null;
  if (isFiniteNumber(input.targetAudienceSize) &&
input.targetAudienceSize > 0) {
    predictedReachRate = safeMul(safeDiv(predictedReach,
input.targetAudienceSize), 100);
  }

  return {
    predictedSpend: newAdSpend,
    predictedImpressions,
    predictedCpm: isFiniteNumber(cpm) ? cpm : null,
    predictedReach,
    predictedFrequency: frequency,
    predictedCostPer1000Reach,
    predictedReachRate,
  };
}

```

Лістинг 3.60 — Воронковий прогноз для Охоплення.

```

function forecastEngagement(input, coeffs) {
  const newAdSpend = preflightFunnel(input);
  if (newAdSpend == null) return null;

  const { cpm, engagementRate, commentRate, shareRate, saveRate } = coeffs;
  if (!isFiniteNumber(cpm) || cpm <= 0) return null;

  const predictedImpressions = safeMul(safeDiv(newAdSpend, cpm), 1000);
  const predictedEngagements = safeMul(predictedImpressions,
pct(engagementRate));
  const predictedCostPerEngagement = safeDiv(newAdSpend,
predictedEngagements);

  return {
    predictedSpend: newAdSpend,
    predictedImpressions,
    predictedEngagementRate: isFiniteNumber(engagementRate) ?
engagementRate : null,
    predictedEngagements,
    predictedCostPerEngagement,
    predictedComments: isFiniteNumber(commentRate)
      ? safeMul(predictedImpressions, pct(commentRate))
      : null,
    predictedShares: isFiniteNumber(shareRate)
      ? safeMul(predictedImpressions, pct(shareRate))
      : null,
    predictedSaves: isFiniteNumber(saveRate)
      ? safeMul(predictedImpressions, pct(saveRate))
      : null,
  };
}

```

Лістинг 3.61 — Воронковий прогноз для Залучення.

Після розрахунку базового прогнозу застосунок формує три сценарії: базовий, оптимістичний і песимістичний. Для цього функція `funnelForecastScenarios` застосовує до базових коефіцієнтів відповідні сценарні множники, після чого викликає воронкову функцію для кожного сценарію.

```

export function funnelForecastScenarios({ campaignType, metrics, inputs,
forecastInputs }) {
  const funnel = FUNNELS[campaignType];
  if (!funnel) return null;

  const base = baseCoeffs(campaignType, metrics, inputs);
  const merged = { ...inputs, ...forecastInputs };

  return reconcileScenarioSet(campaignType,
enforceScenarioOrdering(campaignType, {
  base: funnel(merged, applyScenario('base', base)),
  optimistic: funnel(merged, applyScenario('optimistic', base)),
  pessimistic: funnel(merged, applyScenario('pessimistic', base)),
}), ctx);
}

```

Лістинг 3.62 — Формування сценаріїв воронкового прогнозу

Приклад результату прогнозу через воронку у трьох сценаріях (рис 3. 15).

Прогноз
Простий прогноз через воронку, сценарії та лінійна регресія, якщо доступні історичні дані.

Денний бюджет (Daily Budget) 13000 € Кількість днів прогнозу (Forecast Days) 8

Метод прогнозу
 Прогноз через воронку Прогноз на основі лінійної регресії

Згенерувати прогноз

ПЕСИМІСТИЧНИЙ		БАЗОВИЙ		ОПТИМІСТИЧНИЙ	
Прогнозовані витрати	104 000 €	Прогнозовані витрати	104 000 €	Прогнозовані витрати	104 000 €
Прогнозовані покази	13 159 827	Прогнозовані покази	14 475 810	Прогнозовані покази	15 237 694
Прогнозовані кліки	236 701	Прогнозовані кліки	289 302	Прогнозовані кліки	334 981
Прогнозовані покупки	2 893	Прогнозовані покупки	3 929	Прогнозовані покупки	5 004
Прогнозований дохід	288 585,82 €	Прогнозований дохід	412 533,33 €	Прогнозований дохід	551 709,05 €
Прогнозований ROAS	2,77	Прогнозований ROAS	3,97	Прогнозований ROAS	5,30
Прогнозований CPA	35,95 €	Прогнозований CPA	26,47 €	Прогнозований CPA	20,78 €
Прогнозований прибуток	-2 994,96 €	Прогнозований прибуток	40 386,67 €	Прогнозований прибуток	89 098,17 €

Прогноз є приблизним і базується на припущенні, що ключові показники кампанії залишаються відносно стабільними. Результати реклами можуть змінюватися через аукціон, конкуренцію, сезонність, поведінку користувачів і зміни алгоритмів рекламної платформи.

Рисунок 3.15 — Результати прогнозу через рекламну воронку

3.5.3 Прогноз на основі лінійної регресії

Прогноз на основі лінійної регресії використовується тоді, коли система має достатньо історичних даних: денну деталізацію кампанії, кілька періодів або кілька кампаній одного типу. На відміну від воронкового прогнозу, регресія не просто переносить поточні коефіцієнти на новий бюджет, а намагається визначити залежність між витратами на рекламу та результатами кампанії.

Вибір даних для регресії реалізовано у компоненті `src/components/forecast.js` через функцію `selectRegressionDataset`. Якщо аналізуються періоди, система спочатку намагається використати денні дані з усіх періодів. Якщо денних даних немає, використовуються агреговані значення періодів. Якщо аналізуються кампанії, система може використати денні дані активної кампанії або агреговані значення кількох кампаній.

```

function selectRegressionDataset(analysis, selectedCampaign = null) {
  if (analysis.task === ANALYSIS_TASK.PERIODS && analysis.campaigns.length
  >= 2) {
    const dailyObservations = collectPeriodDailyObservations(analysis);
    if (dailyObservations.length >= 2) {
      return {
        observations: dailyObservations,
        current: analysis.campaigns[analysis.campaigns.length - 1],
        isDaily: true,
      };
    }
    return {
      observations: sortPeriodCampaigns(analysis.campaigns),
      current: analysis.campaigns[analysis.campaigns.length - 1],
      isDaily: false,
    };
  }
  const ownDaily = selectedCampaign?.daily?.length >= 2 ?
  selectedCampaign.daily : null;
  if (ownDaily) {
    return {
      observations: sortDailyRows(ownDaily),
      current: selectedCampaign,
      isDaily: true,
    };
  }

  if (analysis.task === ANALYSIS_TASK.CAMPAIGNS &&
  analysis.campaigns.length >= 2) {
    return {
      observations: sortCampaigns(analysis.campaigns),
      current: selectedCampaign ??
  analysis.campaigns[analysis.campaigns.length - 1],
      isDaily: false,
    };
  }

  return null;
}

```

Лістинг 3.63 — Вибір набору даних для регресійного прогнозу

Сам розрахунок регресійної моделі виконується у файлі `src/calc/statistics.js`.

```
export function linearRegression(xs, ys) {
  const pairs = cleanPairs(xs, ys);
  if (pairs.length < 2) return null;

  const mx = pairs.reduce((s, [x]) => s + x, 0) / pairs.length;
  const my = pairs.reduce((s, [, y]) => s + y, 0) / pairs.length;

  let num = 0;
  let den = 0;

  for (const [x, y] of pairs) {
    const dx = x - mx;
    num += dx * (y - my);
    den += dx * dx;
  }

  if (den === 0) return null;

  const a = num / den;
  const b = my - a * mx;

  return { a, b, n: pairs.length, rse };
}
```

Лістинг 3.64 — Розрахунок лінійної регресії

У повній реалізації функція також розраховує похибку моделі `rse`, яка використовується для сценарного прогнозу. Якщо кількість точок більша за 2, `rse` обчислюється на основі залишків моделі. Якщо доступно лише дві точки, модель може побудувати пряму, але похибка не є повноцінною, тому в інтерфейсі користувач бачить попередження про мінімальну кількість даних.

У модулі `src/calc/forecast.js` регресійний прогноз запускається функцією `regressionForecastScenarios`. Вона перевіряє наявність даних, формує контекст

прогнозу, викликає прямий регресійний прогноз, а потім узгоджує отримані значення з похідними метриками.

```

export function regressionForecastScenarios({
  campaignType,
  observations,
  current,
  forecastInputs,
  isDaily = false,
}) {
  const newAdSpend = preflightFunnel(forecastInputs);

  if (
    newAdSpend == null ||
    !current ||
    !Array.isArray(observations) ||
    observations.length < 2
  ) {
    return null;
  }

  const ctx = {
    campaignType,
    observations,
    current,
    forecastInputs,
    isDaily,
    horizon: Math.max(1, Number(forecastInputs.forecastDays) || 1),
    newAdSpend,
    models: {},
    rses: {},
  };

  const baseRaw = directRegressionForecast(ctx);
  if (!baseRaw || Object.keys(ctx.models).length === 0) return null;
}

```

Лістинг 3.65 — Запуск регресійного прогнозу

Основна функція, яка будує залежність між бюджетом і конкретною метрикою, - `predictBudgetMetric`. Вона бере історичні спостереження, формує масив значень `x` для рекламних витрат і масив `y` для потрібної метрики. Після цього будується регресійна модель і виконується прогноз для нового бюджету.

```
function predictBudgetMetric(ctx, metricId, predictedId, opts = {}) {
  const model = buildRegression(
    ctx.observations,
    (observation) => observationValue(observation, metricId),
    (observation) => observationValue(observation, 'adSpend'),
  );

  if (!model) {
    return {
      value: ctx.fallbackForecast[predictedId] ?? null,
      rse: null,
    };
  }

  const xTarget = ctx.isDaily
    ? Number(ctx.forecastInputs.newDailyBudget)
    : ctx.newAdSpend;

  let value = model.predict(xTarget);
  if (ctx.isDaily) value *= ctx.horizon;

  return {
    value: sanitizePrediction(value,
ctx.fallbackForecast[predictedId], opts), rse };
}
```

Лістинг 3.66 — Прогноз метрики залежно від бюджету

Після прогнозування основних результатів застосунок перераховує похідні метрики.

```

function reconcileSalesRegression(forecast, ctx) {
  const out = { ...forecast };
  const spend = out.predictedSpend;
  const revenue = out.predictedRevenue;
  const purchases = out.predictedPurchases;
  const margin = resolveSalesMarginPct(ctx);

  let grossProfit = null;
  if (isFiniteNumber(revenue) && isFiniteNumber(margin)) {
    grossProfit = safeMul(revenue, margin / 100);
  }

  out.predictedGrossProfit = grossProfit;
  out.predictedProfit =
    isFiniteNumber(grossProfit) && isFiniteNumber(spend) ? grossProfit -
    spend : null;
  out.predictedRoas = safeDiv(revenue, spend);
  out.predictedCpa = safeDiv(spend, purchases);

  return out;
}

```

Лістинг 3.67 — Узгодження похідних метрик для прогнозу Продажі

Якщо регресійна модель для певної метрики не може бути побудована, застосунок використовує резервний воронковий прогноз для цього показника. Це реалізовано через `fallbackForecast`.

Приклад результату прогнозу на основі лінійної регресії у трьох сценаріїв (рис 3. 16).

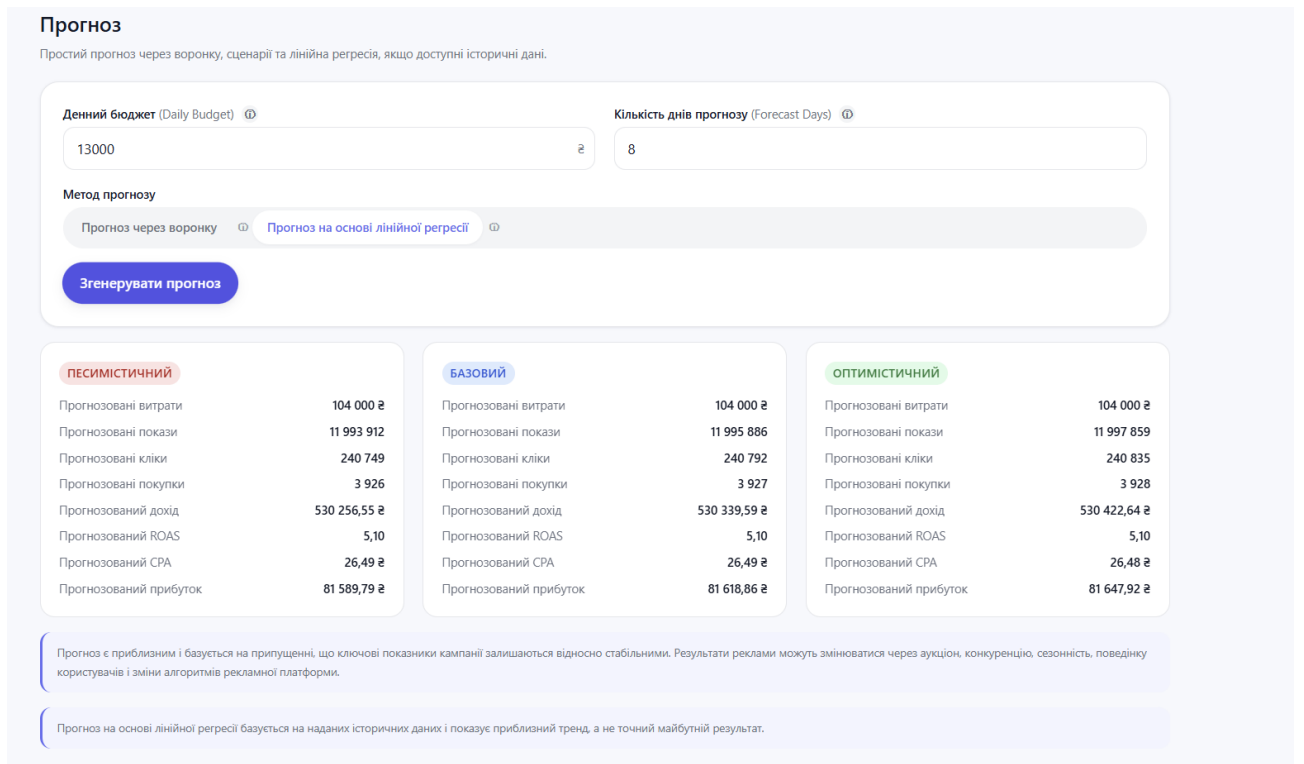


Рисунок 3.16 — Результати прогнозу на основі лінійної регресії

3.5.4 Сценарний прогноз

Щоб прогноз був більш практичним, у застосунку реалізовано сценарний підхід. Замість одного значення система формує три варіанти прогнозу: песимістичний, базовий та оптимістичний. Це дозволяє користувачу побачити не тільки очікуваний результат, а й приблизний діапазон можливих змін за умови погіршення або покращення ефективності кампанії.

Для воронкового прогнозу сценарії формуються через зміну коефіцієнтів рекламної воронки. Сценарні множники зберігаються у файлі `src/config/scenarios.js` у константі `SCENARIO_FACTORS`.

```
export const SCENARIO_FACTORS = Object.freeze({
  optimistic: Object.freeze({
    ctr: 1.1,
    cr: 1.1,
    leadCr: 1.1,
    lpvRate: 1.1,
    engagementRate: 1.1,
    qualifiedLeadRate: 1.1,
    cpm: 0.95,
    cpc: 0.95,
    cpa: 0.95,
    cpl: 0.95,
    costPerLpv: 0.95,
    costPerEngagement: 0.95,
    costPer1000Reach: 0.95,
    costPerQualifiedLead: 0.95,
    aov: 1.05,
    frequency: 1.0,
  }),
  pessimistic: Object.freeze({
    ctr: 0.9,
    cr: 0.9,
    leadCr: 0.9,
    lpvRate: 0.9,
    engagementRate: 0.9,
    qualifiedLeadRate: 0.9,
    cpm: 1.1,
    cpc: 1.1,
    cpa: 1.1,
    cpl: 1.1,
    costPerLpv: 1.1,
    costPerEngagement: 1.1,
    costPer1000Reach: 1.1,
    costPerQualifiedLead:
```

Лістинг 3.68 — Сценарні коефіцієнти для воронкового прогнозу

Застосування сценарію до базових коефіцієнтів реалізовано функцією `applyScenario` у файлі `src/calc/forecast.js`.

```

export function applyScenario(scenario, base) {
  if (scenario === 'base') return normalizeScenarioCoeffs(base);

  const factors = SCENARIO_FACTORS[scenario];
  if (!factors) return normalizeScenarioCoeffs(base);

  const out = {};
  for (const key of Object.keys(base)) {
    const factor = factors[key];

    if (isFiniteNumber(base[key]) && isFiniteNumber(factor)) {
      out[key] = normalizeCoeff(key, base[key] * factor);
    } else {
      out[key] = normalizeCoeff(key, base[key]);
    }
  }

  return out;
}

```

Лістинг 3.69 — Застосування сценарного коефіцієнта

Після зміни коефіцієнтів функція `funnelForecastScenarios` викликає воронковий прогноз окремо для базового, оптимістичного та песимістичного сценаріїв.

```

return reconcileScenarioSet(campaignType,
enforceScenarioOrdering(campaignType, {
  base: funnel(merged, applyScenario('base', base)),
  optimistic: funnel(merged, applyScenario('optimistic', base)),
  pessimistic: funnel(merged, applyScenario('pessimistic', base)),
}), ctx);

```

Лістинг 3.70 — Формування сценаріїв для воронкового прогнозу

Для регресійного прогнозу сценарії формуються інакше. Спочатку система будує базовий прогноз за лінійною регресією. Після цього для показників, для яких доступна похибка моделі RSE, оптимістичний сценарій отримується через додавання цієї похибки до базового значення, а песимістичний через її віднімання.

```
function regressionScenarioForecast(base, scenario, ctx) {
  if (scenario === 'base') return { ...base };

  const out = { ...base };
  const rses = ctx.rses || {};

  for (const [metricId, value] of Object.entries(base)) {
    const rse = rses[metricId];
    if (!isFiniteNumber(rse) || !isFiniteNumber(value)) continue;

    let scenarioValue;
    if (scenario === 'optimistic') {
      scenarioValue = value + rse;
    } else if (scenario === 'pessimistic') {
      scenarioValue = value - rse;
    } else {
      scenarioValue = value;
    }

    const allowNegative = metricId === 'predictedProfit' || metricId ===
'predictedGrossProfit';
    if (!allowNegative && scenarioValue < 0) {
      scenarioValue = 0;
    }

    out[metricId] = scenarioValue;
  }

  return out;
}
```

Лістинг 3.71 — Формування сценаріїв регресійного прогнозу

Для більшості прогнозних показників від’ємні значення не допускаються, тому якщо сценарне значення стає меншим за нуль, воно замінюється на 0. Винятком є `predictedProfit` і `predictedGrossProfit`, оскільки прибуток може бути від’ємним.

Після формування сценаріїв застосунок додатково перевіряє їхній порядок. Для показників, де більше значення є кращим, оптимістичний сценарій має бути не нижчим за базовий, а песимістичний — не вищим. Для вартісних показників, де нижче значення є кращим, логіка протилежна. Це реалізовано функцією `enforceScenarioOrdering`.

```
if (direction === 'higher') {
  out.optimistic[metricId] = Math.max(base, optimistic, pessimistic);
  out.pessimistic[metricId] = Math.min(base, optimistic, pessimistic);
} else if (direction === 'lower') {
  out.optimistic[metricId] = Math.min(base, optimistic, pessimistic);
  out.pessimistic[metricId] = Math.max(base, optimistic, pessimistic);
}
```

Лістинг 3.72 — Формування сценаріїв регресійного прогнозу

3.6 Реалізація візуалізації результатів за допомогою графіків

Для наочного подання результатів аналізу у вебзастосунку реалізовано модуль графіків. Він дозволяє користувачу швидше оцінити ефективність кампанії, порівняти фактичні значення з `benchmark`-показниками, проаналізувати динаміку за днями, порівняти кампанії або періоди, а також переглянути сценарії прогнозу.

Побудова графіків реалізована за допомогою бібліотеки `Chart.js`. Основна логіка графіків знаходиться у файлі `src/components/charts.js`, а загальний блок,

який визначає, які саме графіки потрібно показати, реалізовано у файлі `src/components/charts-block.js`.

На початку файлу `charts.js` бібліотека `Chart.js` підключається та реєструється разом з усіма стандартними елементами. Також у цьому файлі задаються базові параметри стилю графіків: шрифт, кольори осей, сітки та легенди.

```
import { Chart, registerables } from 'chart.js';

Chart.register(...registerables);

Chart.defaults.font.family = FONT_FAMILY;
Chart.defaults.color = AXIS_COLOR;
Chart.defaults.borderColor = GRID_COLOR;
Chart.defaults.scale.grid.color = GRID_COLOR;
Chart.defaults.scale.ticks.color = AXIS_COLOR;
```

Лістинг 3.73 — Підключення `Chart.js` та базові налаштування графіків

Відображення графіків залежить від того, які дані доступні після аналізу. Це реалізовано у компоненті `ChartsBlock`. Якщо результатів аналізу ще немає, застосунок показує порожній стан. Якщо аналіз виконано, система перевіряє наявність даних для кожного типу графіка і додає тільки ті графіки, які можна побудувати.

```

export function ChartsBlock({ analysis, forecast }) {
  if (!analysis) {
    return StateBlock({
      kind: 'empty',
      title: t('state.empty'),
      hint: t('state.emptyHint'),
    });
  }

  const cards = [];

  if (hasActualVsBenchmarkData(analysis)) {
    cards.push(ActualVsBenchmarkChart({ analysis }).el);
  }

  if (analysis.task === ANALYSIS_TASK.CAMPAIGNS &&
analysis.campaigns.length > 1) {
    cards.push(ByCampaignChart({ analysis }).el);
  }

  if (analysis.task === ANALYSIS_TASK.PERIODS && analysis.campaigns.length
> 1) {
    cards.push(ByPeriodChart({ analysis }).el);
  }

  const hasDaily = analysis.campaigns.some((c) => c.daily &&
c.daily.length > 0);
  if (hasDaily) {
    cards.push(DailyChart({ analysis }).el);
  }
  if (forecast?.results) {
    cards.push(...ScenarioCharts({
      campaignType: analysis.campaignType,
      scenarios: forecast.results,
      currency: analysis.currency,
    }));
  }
  return el('div', { class: 'charts-grid' }, cards);
}

```

Лістинг 3.74 — Формування блоку графіків

Першим типом візуалізації є графік порівняння фактичних значень із benchmark або внутрішньою базою порівняння. Він реалізований у функції ActualVsBenchmarkChart. Для цього графіка використовується відносне порівняння: значення 100% відповідає benchmark або внутрішньому середньому. Якщо фактичне значення вище або нижче бази порівняння, це відображається на стовпчиковому графіку (рис. 3.17).

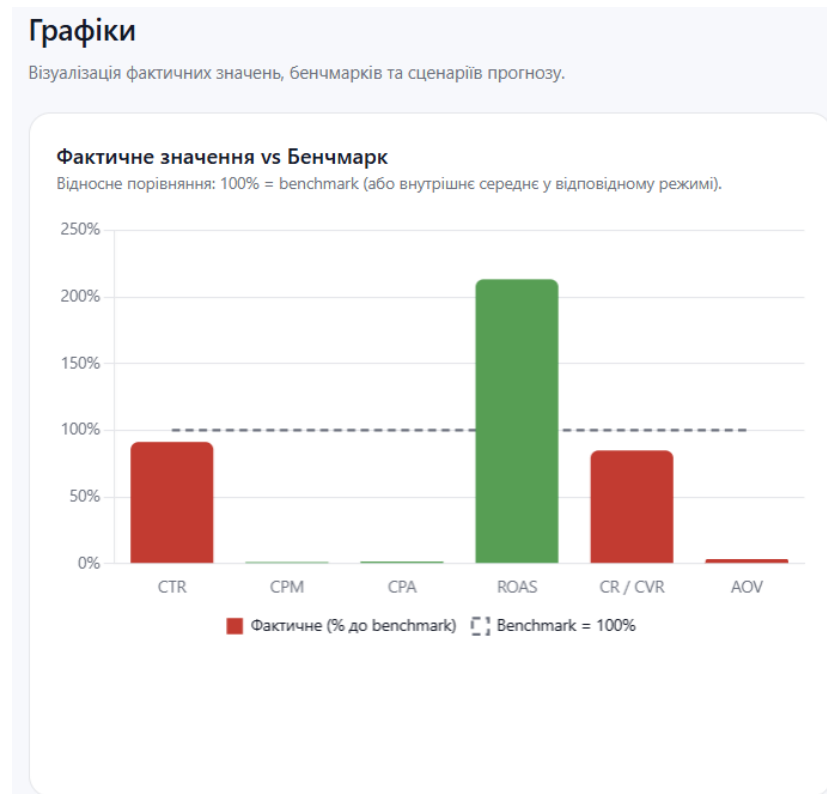


Рисунок 3.17 — Графік порівняння фактичних значень із benchmark-показниками

Для порівняння кількох кампаній використовується графік ByCampaignChart. Він будується тільки тоді, коли користувач обрав задачу порівняння кампаній і в аналізі є більше однієї кампанії. У цьому графіку користувач може вибрати метрику зі списку, а система побудує стовпчикову діаграму за всіма кампаніями (рис 3.18).

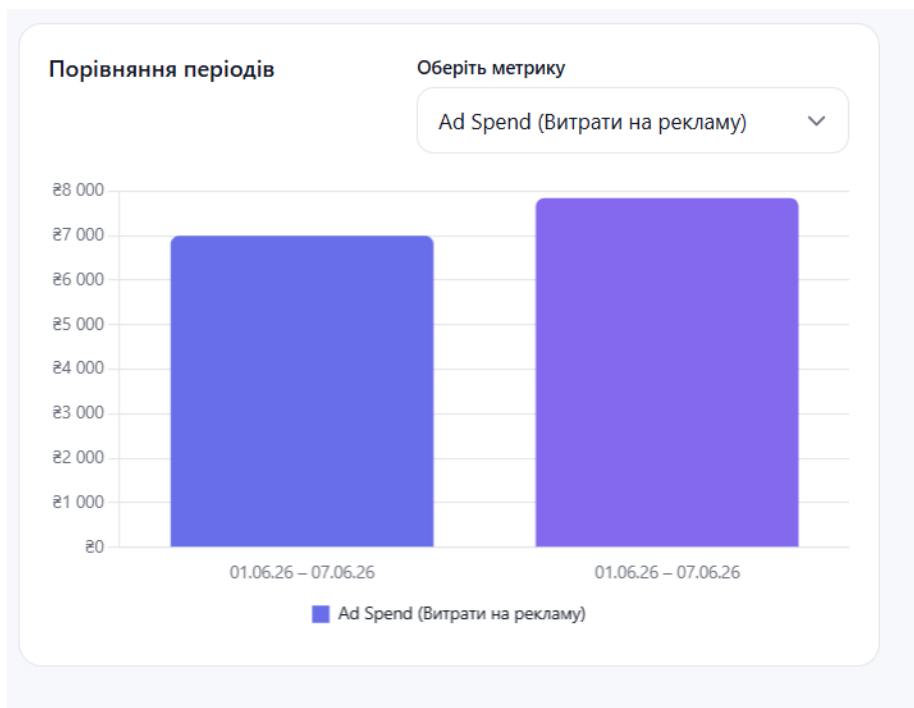


Рисунок 3.18 — Графік порівняння періодів за обраною метрикою

Для порівняння періодів використовується аналогічний графік `VuPeriodChart`. Він дозволяє оцінити, як змінювався обраний показник між різними періодами рекламної кампанії. Набір доступних метрик для таких графіків задається у конфігурації `COMPARISON_METRICS`.

```
const opts = COMPARISON_METRICS[analysis?.campaignType] || [];
const state = { metric: initialMetric || opts[0] };

const select = SelectField({
  label: t('step.metricForChart'),
  value: state.metric,
  options: opts.map((id) => ({ value: id, label: metricLabel(id) })),
  onChange: (v) => {
    state.metric = v;
    refresh();
  },
});
```

Лістинг 3.75 — Вибір метрики для порівняння кампаній

Окремо реалізовано графіки сценаріїв прогнозу. Вони будуються після того, як користувач згенерував прогноз. Для цього використовується функція `ScenarioCharts`, яка отримує результати песимістичного, базового та оптимістичного сценаріїв і будує стовпчикові графіки для груп прогнозних показників.

```
export function ScenarioCharts({ campaignType, scenarios, currency }) {
  const groups = scenarioGroups(campaignType);
  const out = [];

  for (const group of groups) {
    const card = chartCard({ title: group.title, hint: group.subtitle });
    const points = resolveGroupPoints(group, scenarios);

    const labels = points.map((p) => p.label);
    const datasets = SCENARIO_DATASETS.map((s, idx) => ({
      label: s.label,
      data: points.map((p) => p.values[idx]),
      backgroundColor: s.color,
      borderRadius: 6,
    }));

    card.update({
      type: 'bar',
      data: { labels, datasets },
      options: scenarioBarOptions(group.unit, currency),
    });

    out.push(card.el);
  }

  return out;
}
```

Лістинг 3.76 — Побудова графіків сценарного прогнозу

Приклад графіків сценарного прогнозу (рис.3.19).



Рисунок 3.19 — Графіки сценарії прогнозу

Якщо для певного графіка недостатньо даних, застосунок не намагається побудувати некоректну візуалізацію. У такому випадку в тілі картки показується повідомлення про недостатню кількість даних. Це реалізовано через функції `showEmpty` і `hideEmpty`, які керують відображенням `canvas`-елемента та порожнього стану.

РОЗДІЛ 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ

У попередньому розділі було описано практичну реалізацію веб-застосунку: введення даних, обчислення рекламних метрик, benchmark-порівняння, статистичний аналіз, прогнозування та візуалізацію результатів. Метою цього розділу є перевірка того, що розроблений застосунок коректно працює в типових і крайових сценаріях, правильно обробляє вхідні дані, не виконує некоректних обчислень і формує зрозумілі результати для користувача.

Тестування виконувалося з позиції кінцевого користувача, тобто через інтерфейс веб-застосунку. Основна увага приділялася повному шляху роботи з системою: вибір типу аналізу, введення або завантаження даних, перевірка коректності значень, розрахунок метрик, формування інтегральної оцінки, побудова прогнозу, відображення графіків і рекомендацій. Такий підхід дозволяє перевірити не тільки окремі математичні формули, а й узгодженість роботи всіх частин застосунку в одному сценарії.

4.1 Мета та підхід до тестування веб-застосунку

Метою тестування розробленого веб-застосунку є перевірка його працездатності в основних сценаріях використання та підтвердження коректності отриманих аналітичних результатів. У цьому розділі не розглядається внутрішня реалізація програмних модулів, оскільки вона була описана в попередньому розділі. Основна увага приділяється тому, як готовий застосунок працює з позиції користувача: чи правильно він приймає дані, чи

виявляє помилки, чи коректно розраховує метрики та чи зрозуміло відображає результати аналізу.

Тестування виконувалося за функціональним підходом. Це означає, що перевірялися не окремі фрагменти коду, а повні сценарії роботи із системою: вибір задачі аналізу, вибір типу рекламної кампанії, введення або завантаження даних, запуск аналізу, перегляд розрахованих показників, рекомендацій, прогнозу та графіків. Такий підхід дозволяє оцінити роботу застосунку як єдиної системи.

Під час тестування було перевірено три основні групи сценаріїв. Перша група стосується введення даних: ручного заповнення форми та завантаження CSV-файлів. Друга група пов'язана з перевіркою правильності розрахунків рекламних метрик, статистичних показників і Campaign Score. Третя група охоплює перевірку відображення результатів: таблиць, статусів порівняння, рекомендацій, прогнозів і графіків.

Для тестування було використано підготовлений набір вхідних даних для різних типів рекламних кампаній: «Продажі», «Лідогенерація», «Трафік», «Охоплення» та «Залучення». Окремо було підготовлено файли з некоректними значеннями, щоб перевірити, чи здатна система виявляти помилки до початку розрахунків. Це важливо, оскільки неправильні або суперечливі вхідні дані можуть призвести до хибних аналітичних висновків.

Таким чином, тестування було спрямоване не лише на перевірку математичної правильності формул, а й на оцінювання практичної надійності застосунку. Система повинна не тільки виконувати розрахунки, але й запобігати некоректному використанню, пояснювати обмеження даних і подавати результати у зрозумілій для користувача формі.

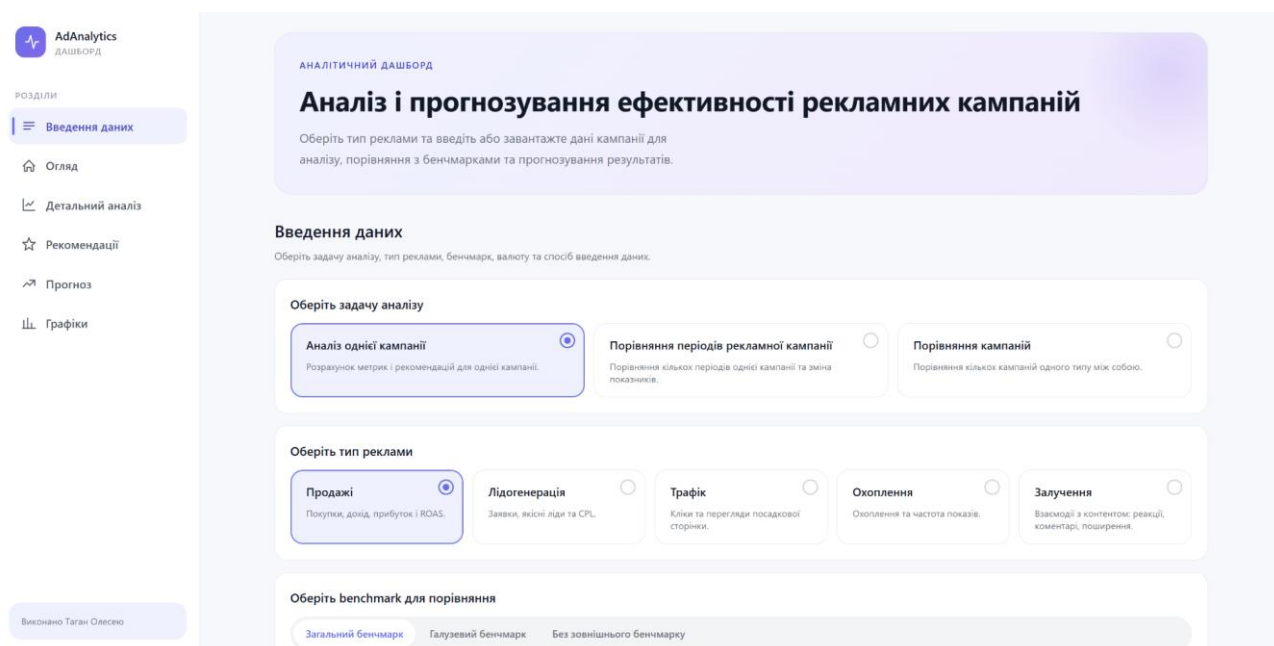


Рисунок 4.1 — Початковий екран тестування веб-застосунку

4.2 Тестування введення та перевірки даних

Першим етапом тестування була перевірка введення даних, оскільки саме від коректності початкових значень залежить правильність усіх подальших розрахунків. У застосунку передбачено два способи введення даних: ручне заповнення форми та завантаження CSV-файлу. Під час тестування було перевірено обидва варіанти.

Спочатку перевірялося ручне введення даних. Для цього було обрано задачу «Аналіз однієї кампанії» та тип реклами «Продажі». У форму були введені коректні значення витрат на рекламу, доходу, кількості покупок, показів, кліків, охоплення та середньої валової маржі. Після заповнення всіх обов'язкових полів застосунок дозволив запуснути аналіз і перейшов до відображення результатів

Введення даних

Назва кампанії (Campaign Name) ①: Test 1

Дата початку (Start Date) ①: 01.06.2026

Дата завершення (End Date) ①: 07.06.2026

Витрати на рекламу (Ad Spend) ①: 2000

Дохід (Revenue) ①: 8000

Покупки (Purchases) ①: 10

Покази (Impressions) ①: 25000

Кліки (Clicks) ①: 500

Охоплення (Reach) ① — необов'язкове: 0

Частота (Frequency) ① — необов'язкове: 0,0

Денний бюджет (Daily Budget) ① — необов'язкове: 0,00

Метод розрахунку прибутковості: Через повну собівартість Через середню валову маржу

Загальна собівартість товарів (Total Product Cost) ①: 1500

Рисунок 4.2 — Коректно заповнена форма для кампанії Продажі

Далі перевірялася поведінка системи при некоректному введенні. У тестовому сценарії кількість кліків була навмисно встановлена більшою за кількість показів. Така ситуація є логічно неможливою, оскільки користувач не може здійснити більше кліків, ніж було показів рекламного оголошення. У результаті застосунок відобразив повідомлення про помилку біля відповідного поля та не дозволив виконати аналіз.

Також було перевірено інші помилкові сценарії: порожні обов'язкові поля, від'ємні числові значення та некоректні співвідношення між показниками. У всіх випадках система не запускала розрахунки до виправлення даних. Це підтверджує, що перевірка значень виконується до етапу аналізу, а не після отримання помилкового результату.

Введення даних

Назва кампанії (Campaign Name) ⓘ Invalid Clicks Test	Дата початку (Start Date) ⓘ 01.06.2026	Дата завершення (End Date) ⓘ 07.06.2026	Витрати на рекламу (Ad Spend) ⓘ 500
Дохід (Revenue) ⓘ 3200	Покупки (Purchases) ⓘ 8	Покази (Impressions) ⓘ 10000	Кліки (Clicks) ⓘ 12000
Охоплення (Reach) ⓘ — необов'язкове 6000	Частота (Frequency) ⓘ — необов'язкове 0,0	Денний бюджет (Daily Budget) ⓘ — необов'язкове 0,00	Метод розрахунку прибутковості Через повну собівартість <input type="radio"/> Через середню валову маржу <input checked="" type="radio"/>
Середня валова маржа % (Average Gross Margin %) ⓘ 35 %			

Перевірте введені значення.


- Кількість кліків не може бути більшою за кількість показів.

Рисунок 4.3 — Приклад помилки введення некоректних даних

Після ручного введення було протестовано завантаження CSV-файлів. Для коректного сценарію використовувався файл із щоденною розбивкою рекламної кампанії. Після завантаження застосунок успішно обробив файл, визначив кількість рядків і підготував дані до аналізу. Наявність щоденної розбивки дозволила системі не тільки розрахувати основні метрики, а й надалі виконати аналіз динаміки.

Метод розрахунку прибутковості
Метод розрахунку прибутковості буде визначено автоматично після завантаження CSV.
Метод розрахунку прибутковості визначено за CSV: Через середню валову маржу.

Перетягніть CSV-файл сюди


Перетягніть CSV-файл сюди
 або натисніть, щоб обрати файл. Перший рядок повинен містити назви колонок.

Усі грошові значення у CSV мають бути в одній вибраній валюті. Конвертація не виконується.

Файл оброблено: 1 рядків, 1 кампанія. Без щоденної розбивки — детальна аналітика обмежена.

Рисунок 4.4 — Приклад успішного завантаження CSV-файлу

Окремо перевірялося завантаження CSV-файлів із помилками. Наприклад, для Sales-кампанії було протестовано файл без обов'язкового поля доходу, а для Traffic-кампанії - файл, у якому кількість переглядів посадкової сторінки перевищувала кількість кліків. У таких випадках застосунок не запускав аналіз і показував повідомлення про проблему. Це дає змогу користувачу виправити дані до початку розрахунків.

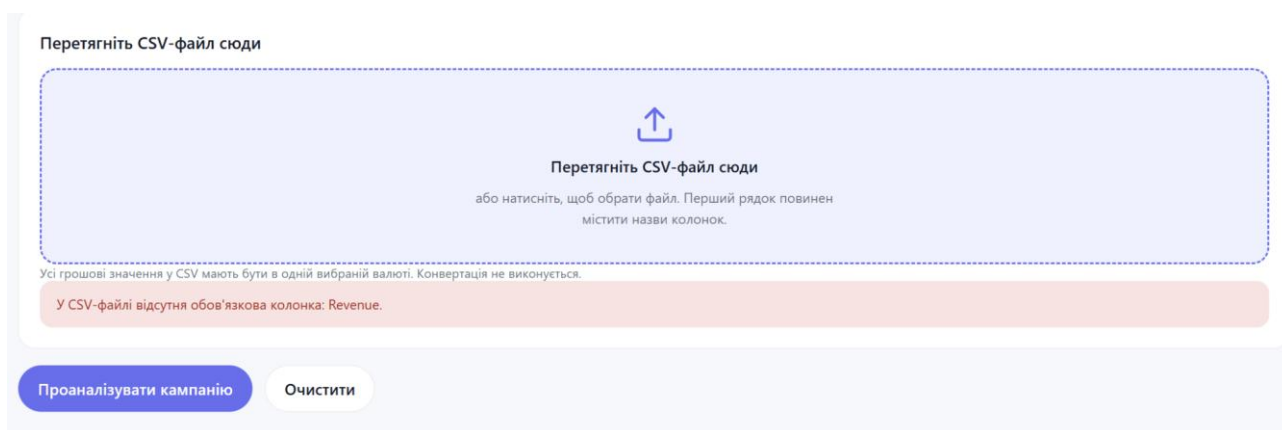


Рисунок 4.5 — Помилка при завантаженні некоректного CSV-файлу

4.3 Перевірка правильності розрахунку рекламних метрик

Наступним етапом тестування була перевірка правильності розрахунку рекламних метрик. Для цього використовувався тестовий CSV-файл із коректними даними Sales-кампанії, оскільки для цього типу кампанії система розраховує найбільшу кількість показників: ROAS, CPA, CTR, CPC, CPM, CR, AOV, валовий прибуток, чистий прибуток та інші економічні метрики.

Для перевірки було використано файл `figure_4_4_sales_valid_daily.csv`, який містить щоденні дані кампанії за п'ять днів. Після завантаження файлу

система агрегує значення за всі дні та виконує розрахунок підсумкових метрик кампанії.

Сумарні вхідні дані для тестового сценарію:

- Витрати на рекламу: 3500
- Дохід: 19200
- Покупки: 48
- Покази: 70000
- Кліки: 1705
- Охоплення: 32200
- Середня валова маржа: 35 %

На основі цих значень були виконані контрольні ручні розрахунки.

Показник ROAS розраховується як відношення доходу до витрат на рекламу:

$$\text{ROAS} = 19200 / 3500 = 5,49$$

$$\text{CPA} = 3500 / 48 = 72,92$$

$$\text{CTR} = 1705 / 70000 * 100 = 2,44 \%$$

$$\text{CPC} = 3500 / 1705 = 2,05$$

$$\text{CPM} = 3500 / 70000 * 1000 = 50,00$$

$$\text{CR} = 48 / 1705 * 100 = 2,82 \%$$

$$\text{AOV} = 19200 / 48 = 400,00$$

Оскільки у тестовому сценарії використовується метод розрахунку прибутковості через середню валову маржу, додатково було перевірено розрахунок прибутку. При маржі 35 % валовий прибуток становить:

$$\text{Валовий прибуток} = 19200 * 0,35 = 6720$$

Також було перевірено розрахунок беззбиткового ROAS:

$$\text{Break-even ROAS} = 1 / 0,35 = 2,86$$

Після запуску аналізу значення, отримані в інтерфейсі застосунку, були порівняні з ручними розрахунками. Результати збіглися з урахуванням округлення, що підтверджує коректність розрахунку основних рекламних метрик для кампанії Продажі.

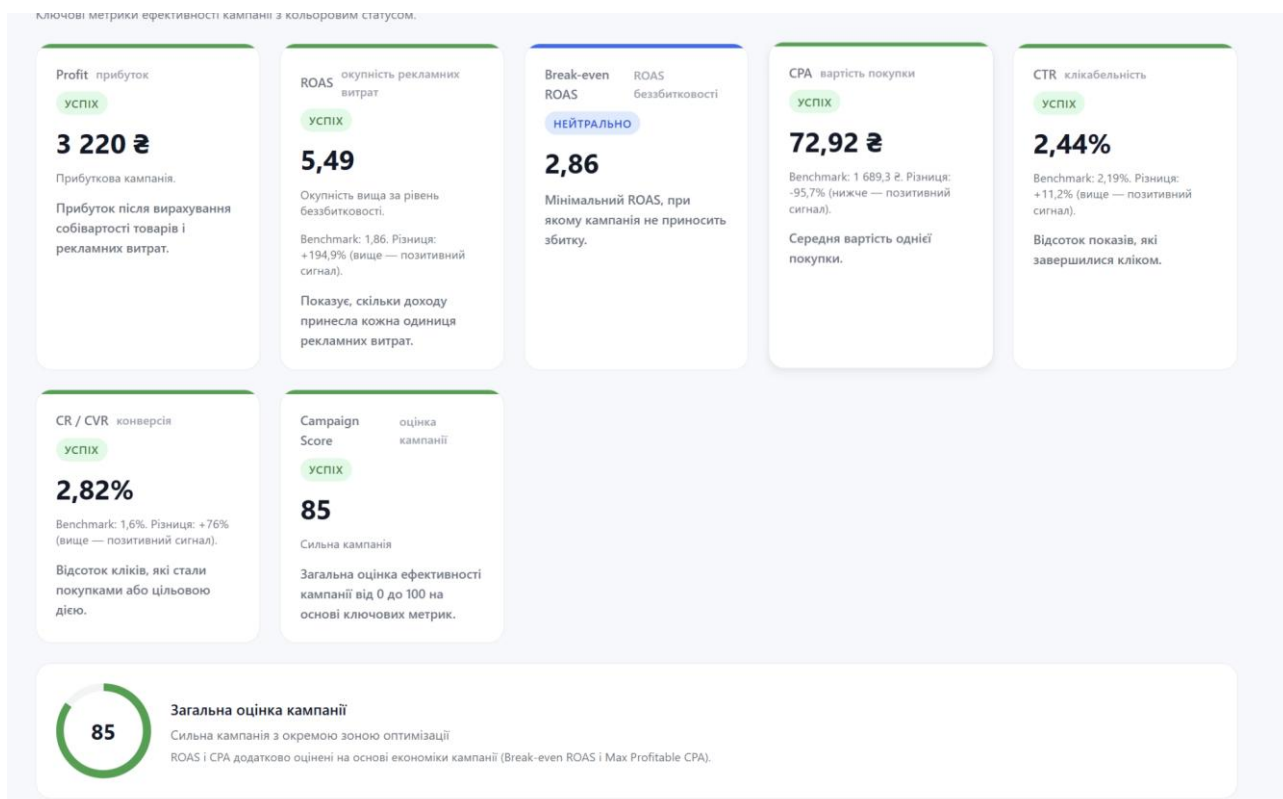


Рисунок 4.6 — Результати розрахунку основних метрик кампанії Продажі

Окрім базових метрик, було перевірено порівняння фактичних значень із benchmark-показниками. У цьому режимі система не лише показує числове значення метрики, а й визначає її статус: показник є кращим, гіршим або близьким до benchmark. Це дозволяє користувачу швидко зрозуміти, які сторони кампанії є сильними, а які потребують оптимізації.

Детальний аналіз

Пояснення метрик, порівняння з бенчмарками та внутрішньою базою порівняння.

Аналіз прибутковості

Кампанія є прибутковою: Profit позитивний, ROAS вищий за Break-even ROAS, а CPA нижчий за Max Profitable CPA.

Duration (тривалість періоду) 5 днів	Ad Spend (Витрати на рекламу) 3 500 ₪	Revenue (Дохід) 19 200 ₪	Gross Profit (валовий прибуток) 6 720 ₪	Gross Margin (валова маржа) 35%	Profit (прибуток) 3 220 ₪	ROAS (окупність рекламних витрат) 5,49
Break-even ROAS (ROAS безбитковості) 2,86	CPA (вартість покупки) 72,92 ₪	Max Profitable CPA (максимально допустимий CPA) 140 ₪	AOV (середній чек) 400 ₪			

Аналіз трафіку

CTR значно вищий за benchmark (позитивний сигнал) і CPM значно нижчий за benchmark (позитивний сигнал). CPC, Frequency показані як фактичні значення без зовнішньої оцінки.

CTR (кликабельність) 2,44%	CPC (вартість кліку) 2,05 ₪	CPM (вартість 1000 показів) 50 ₪	Frequency (Частота) 2,17
--------------------------------------	---------------------------------------	--	------------------------------------

Рисунок 4.7 — Детальний аналіз метрик з порівнянням до benchmark

Аналіз трафіку

CTR значно вищий за benchmark (позитивний сигнал) і CPM значно нижчий за benchmark (позитивний сигнал). CPC, Frequency показані як фактичні значення без зовнішньої оцінки.

CTR (кликабельність) 2,44%	CPC (вартість кліку) 2,05 ₪	CPM (вартість 1000 показів) 50 ₪	Frequency (Частота) 2,17
--------------------------------------	---------------------------------------	--	------------------------------------

Аналіз конверсій

CTR і CR / CVR вищі за базу порівняння. Кампанія працює ефективно на ключових етапах воронки.

CR / CVR (конверсія) 2,82%	CPA (вартість покупки) 72,92 ₪	AOV (середній чек) 400 ₪	Purchases (Покупки) 48
--------------------------------------	--	------------------------------------	----------------------------------

Порівняння з бенчмарками

CTR (кликабельність) 2,44% benchmark: 2,19% (Triple Whale 2026) Різниця до benchmark: +11,2% (вище - позитивний сигнал).	CPC (вартість кліку) 2,05 ₪ Зовнішній benchmark для цієї метрики недоступний. Аналіз виконано на основі доступних даних.	CPM (вартість 1000 показів) 50 ₪ benchmark: 627,68 ₪ (Triple Whale 2026) Різниця до benchmark: -92% (нижче - позитивний сигнал).	CPA (вартість покупки) 72,92 ₪ benchmark: 1 689,3 ₪ (Triple Whale 2026) Різниця до benchmark: -95,7% (нижче - позитивний сигнал).	ROAS (окупність рекламних витрат) 5,49 benchmark: 1,86 (Triple Whale 2026) Різниця до benchmark: +194,9% (вище - позитивний сигнал).	CR / CVR (конверсія) 2,82% benchmark: 1,6% (Triple Whale 2026) Різниця до benchmark: +76% (вище - позитивний сигнал).	AOV (середній чек) 400 ₪ benchmark: 3 171,14 ₪ (Triple Whale 2026) Різниця до benchmark: -87,4% (нижче - негативний сигнал).
--	---	--	---	--	---	--

Рисунок 4.8 — Детальний аналіз метрик з порівнянням до benchmark

4.4 Тестування статистичного аналізу та Campaign Score

Наступним етапом було протестовано роботу статистичного аналізу та інтегральної оцінки Campaign Score. На цьому етапі перевірялося не тільки правильне обчислення окремих метрик, а й те, як система узагальнює результати рекламної кампанії та визначає її загальну ефективність.

Для перевірки статистичного аналізу використовувався CSV-файл із щоденною розбивкою даних `figure_4_4_sales_valid_daily.csv`. Наявність декількох рядків із датами дозволяє системі оцінити зміну показників у часі та виконати аналіз відхилень. Після завантаження файлу застосунок розраховує метрики для кожного дня, порівнює їх із середніми значеннями та визначає, чи є окремі показники помітно вищими або нижчими за загальну динаміку кампанії.

Під час тестування було перевірено, що система не виконує статистичний аналіз у випадках, коли для нього недостатньо даних. Якщо користувач вводить лише одну кампанію без щоденної розбивки, застосунок не формує ненадійні статистичні висновки, а показує повідомлення про обмеженість даних. Така поведінка є правильною, оскільки Z-score та аналіз динаміки мають сенс лише за наявності декількох спостережень.

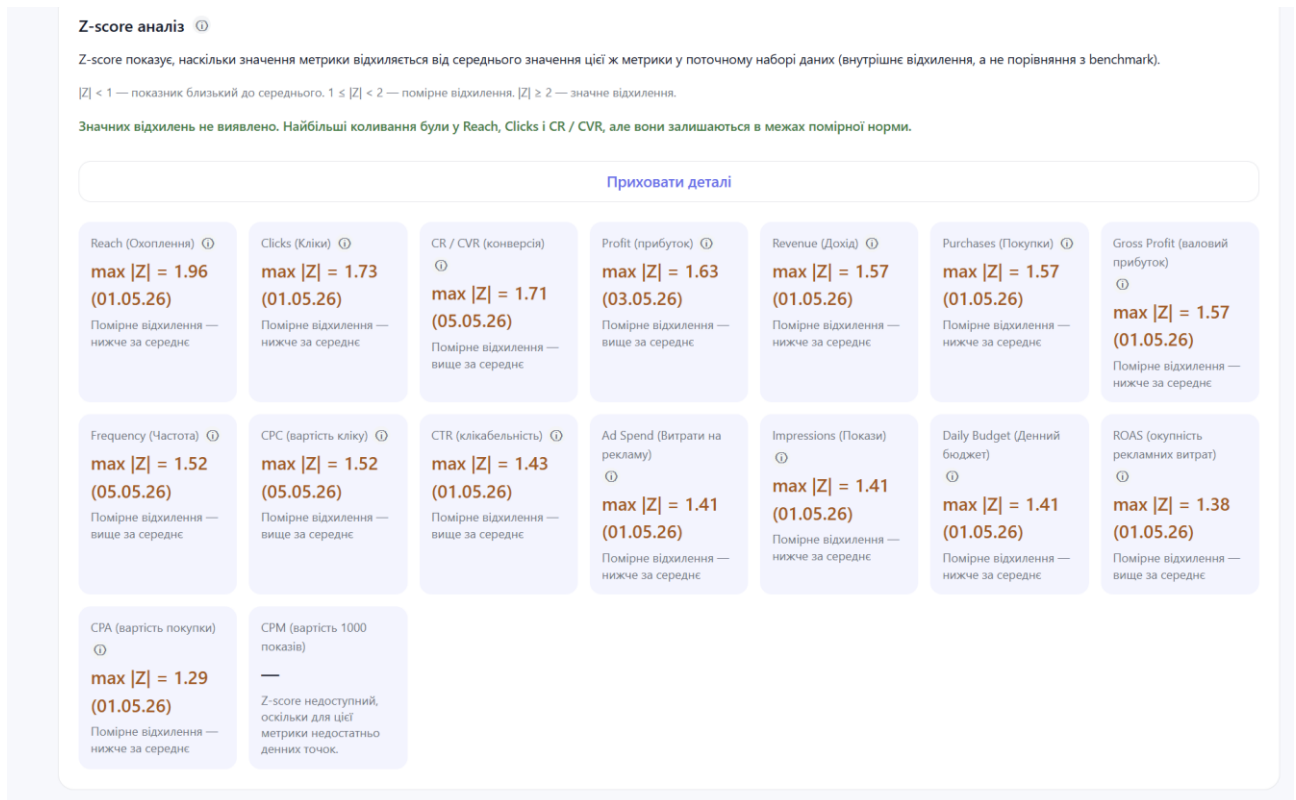


Рисунок 4.9 — Приклад Z-score аналізу за щоденними даними

Окремо було перевірено кореляційний аналіз. Для цього використовувався той самий файл `figure_4_4_sales_valid_daily.csv`, оскільки він містить щоденні значення витрат, доходу, покупок, показів, кліків та інших показників. Завдяки цьому система може порівняти, як різні метрики змінюються відносно одна одної.

Під час тестування було перевірено, що система формує кореляційний аналіз тільки тоді, коли доступна достатня кількість спостережень. Якщо користувач аналізує лише один рядок даних або одну кампанію без динаміки, застосунок не робить статистично ненадійних висновків і повідомляє, що даних для кореляційного аналізу недостатньо.

У тестовому сценарії з щоденними даними очікуваним є позитивний зв'язок між кількістю кліків, покупок і доходом. Це пояснюється тим, що зі збільшенням кількості кліків у тестових даних також зростає кількість покупок і загальний дохід. Водночас результати кореляції не слід трактувати як прямий

причинно-наслідковий зв'язок. Кореляція показує лише статистичну залежність між показниками, а остаточні маркетингові висновки мають враховувати контекст кампанії, аудиторію, креативи та якість посадкової сторінки.

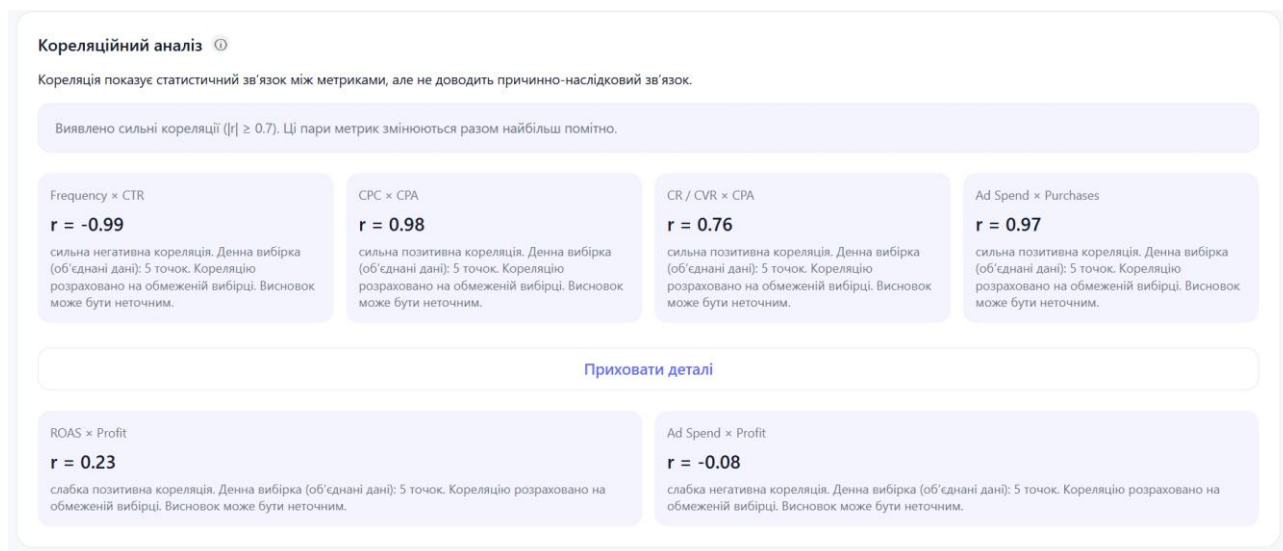


Рисунок 4.10 — Кореляційний аналіз рекламних метрик

Також було протестовано режим порівняння кількох кампаній та періодів. Для цього використовувався сценарій, у якому в систему завантажуються або вводяться дані декількох кампаній Продажі та декілька періодів. У цьому режимі застосунок має порівняти кампанії і періоди між собою за ключовими показниками: витратами, прибутком, ROAS, CPA, CTR, CPC та конверсією. Метою перевірки було переконатися, що система не оцінює кампанії лише за одним показником, а показує загальну картину ефективності.

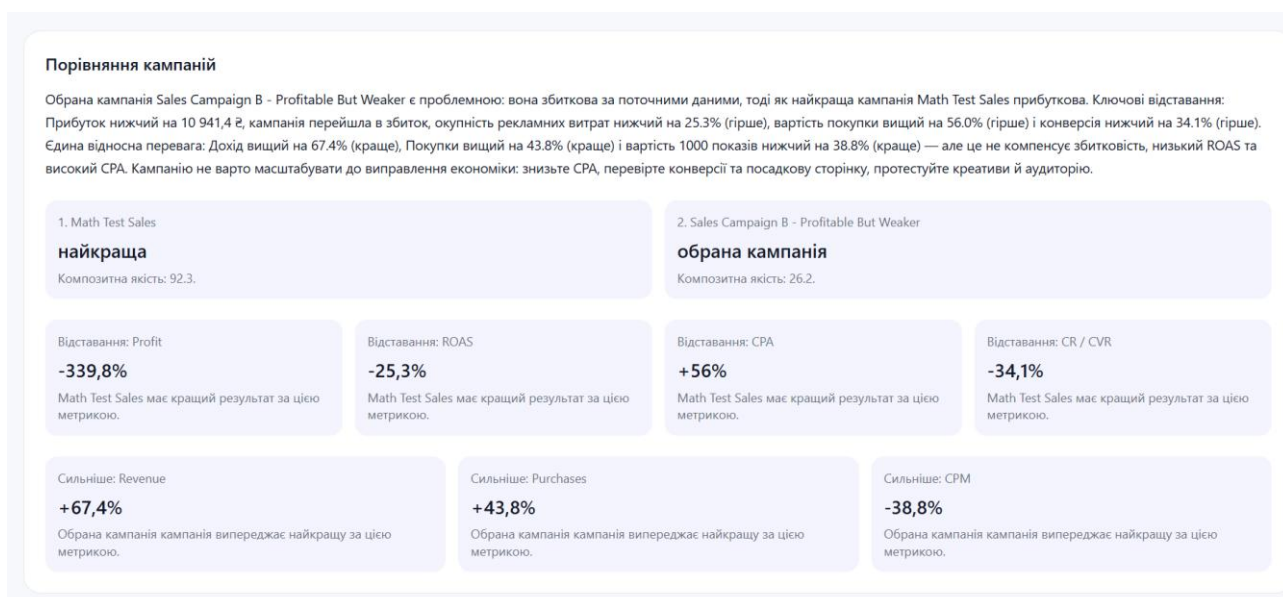


Рисунок 4.11 — Порівняння кількох кампаній

Окремо було перевірено роботу інтегральної оцінки Campaign Score. У тестовому кампанія Продажі має позитивний прибуток, ROAS вищий за беззбитковий рівень і прийнятну вартість покупки. Тому система формує позитивну інтегральну оцінку. Якщо ж у кампанії є слабкі показники, наприклад низька конверсія або зavelика вартість покупки, Campaign Score зменшується. Це підтверджує, що оцінка враховує баланс між різними метриками.

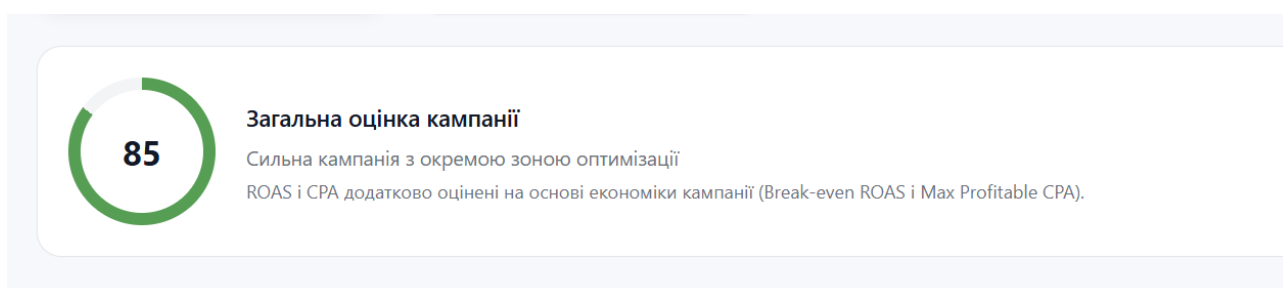


Рисунок 4.12 — Віджет Campaign Score після аналізу кампанії

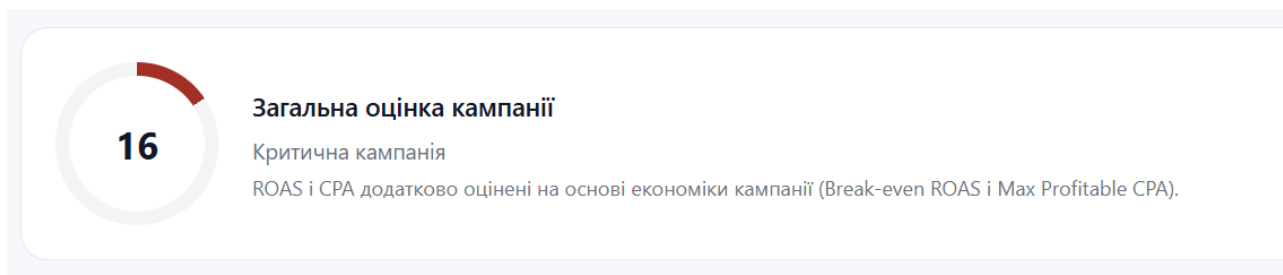


Рисунок 4.13 — Віджет Campaign Score після аналізу кампанії

Отже, тестування показало, що статистичний аналіз, кореляційний аналіз і Campaign Score працюють як додатковий рівень інтерпретації результатів. Вони не замінюють окремі метрики, а допомагають користувачу швидше зрозуміти загальний стан рекламної кампанії, побачити зв'язки між показниками та визначити, чи є результати стабільними, сильними або проблемними.

4.5 Тестування модуля прогнозування

Наступним етапом було протестовано модуль прогнозування результатів рекламної кампанії. Метою цього тестування є перевірка того, чи може застосунок на основі вже проаналізованих даних побудувати орієнтовний прогноз майбутніх результатів при зміні рекламного бюджету.

Прогнозування перевірялося після завантаження та аналізу файлу `figure_4_4_sales_valid_daily.csv`. Спочатку система розраховує фактичні показники кампанії, а вже після цього користувач переходить до блоку «Прогноз». Такий порядок є правильним, оскільки прогноз має базуватися не на довільних числах, а на вже перевірених і розрахованих показниках кампанії.

Для тестування було використано такі параметри прогнозу:

- Новий денний бюджет: 1000
- Кількість днів прогнозу: 7

— Метод прогнозу: Прогноз через воронку

Після введення цих значень застосунок формує прогноз майбутніх результатів кампанії. Для Sales-кампанії система розраховує прогнозовані покази, кліки, покупки, дохід, ROAS, CPA та прибуток. Перевірялося, що при збільшенні бюджету прогнозовані значення змінюються логічно: зростають витрати, кількість показів, кліків і покупок, а економічні показники залишаються узгодженими з поточними коефіцієнтами кампанії.

Прогноз

Простий прогноз через воронку, сценарії та лінійна регресія, якщо доступні історичні дані.

Денний бюджет (Daily Budget) 1000

Кількість днів прогнозу (Forecast Days) 7

Метод прогнозу

Прогноз через воронку Прогноз на основі лінійної регресії

Згенерувати прогноз

Рисунок 4.14 — Налаштування прогнозу після виконаного аналізу

Окремо було перевірено сценарний прогноз. Застосунок формує три варіанти майбутнього результату: песимістичний, базовий та оптимістичний. Песимістичний сценарій показує можливий результат при погіршенні частини коефіцієнтів кампанії, базовий сценарій спирається на поточні показники, а оптимістичний демонструє результат за умови покращення ефективності. Під час тестування перевірялося, що сценарії розташовані логічно. Для показників, де більше значення є кращим, наприклад дохід, покупки або прибуток, оптимістичний сценарій має бути вищим за базовий, а базовий - вищим за песимістичний. Для показників, де менше значення є кращим, наприклад CPA, навпаки, оптимістичний сценарій має показувати нижчу вартість покупки.

ПЕСИМІСТИЧНИЙ		БАЗОВИЙ		ОПТИМІСТИЧНИЙ	
Прогнозовані витрати	7 000 €	Прогнозовані витрати	7 000 €	Прогнозовані витрати	7 000 €
Прогнозовані покази	127 273	Прогнозовані покази	140 000	Прогнозовані покази	147 368
Прогнозовані кліки	2 790	Прогнозовані кліки	3 410	Прогнозовані кліки	3 948
Прогнозовані покупки	71	Прогнозовані покупки	96	Прогнозовані покупки	122
Прогнозований дохід	26 862,55 €	Прогнозований дохід	38 400 €	Прогнозований дохід	51 354,95 €
Прогнозований ROAS	3,84	Прогнозований ROAS	5,49	Прогнозований ROAS	7,34
Прогнозований CPA	99,02 €	Прогнозований CPA	72,92 €	Прогнозований CPA	57,25 €
Прогнозований прибуток	2 401,89 €	Прогнозований прибуток	6 440 €	Прогнозований прибуток	10 974,23 €

Прогноз є приблизним і базується на припущенні, що ключові показники кампанії залишаються відносно стабільними. Результати реклами можуть змінюватися через аукціон, конкуренцію, сезонність, поведінку користувачів і зміни алгоритмів рекламної платформи.

Рисунок 4.15 — Результати сценарного прогнозу через воронку кампанії Продажі

Також було протестовано прогноз на основі лінійної регресії. Цей метод доступний тоді, коли у системи є історичні дані, наприклад щоденна розбивка кампанії. У такому випадку застосунок може не лише використати поточні коефіцієнти рекламної воронки, а й врахувати тенденцію зміни показників у часі.

Для перевірки регресійного прогнозу використовувався той самий файл `figure_4_4_sales_valid_daily.csv`, оскільки він містить дані за п'ять днів. У блоці прогнозу було обрано метод «Прогноз на основі лінійної регресії». Після запуску прогнозу система побудувала результат на основі історичних точок і показала орієнтовні значення для майбутнього періоду.

- Новий денний бюджет: 1000.
- Кількість днів прогнозу: 7.
- Метод прогнозу: Прогноз на основі лінійної регресії.

Прогноз
Простий прогноз через воронку, сценарії та лінійна регресія, якщо доступні історичні дані.

Денний бюджет (Daily Budget) 1000 € Кількість днів прогнозу (Forecast Days) 7

Метод прогнозу
Прогноз через воронку **Прогноз на основі лінійної регресії**

Згенерувати прогноз

ПЕСИМІСТИЧНИЙ		БАЗОВИЙ		ОПТИМІСТИЧНИЙ	
Прогнозовані витрати	7 000 €	Прогнозовані витрати	7 000 €	Прогнозовані витрати	7 000 €
Прогнозовані покази	140 000	Прогнозовані покази	140 000	Прогнозовані покази	140 000
Прогнозовані кліки	2 658	Прогнозовані кліки	2 723	Прогнозовані кліки	2 788
Прогнозовані покупки	80	Прогнозовані покупки	82	Прогнозовані покупки	84
Прогнозований дохід	31 874,56 €	Прогнозований дохід	32 760 €	Прогнозований дохід	33 645,44 €
Прогнозований ROAS	4,55	Прогнозований ROAS	4,68	Прогнозований ROAS	4,81
Прогнозований CPA	87,84 €	Прогнозований CPA	85,47 €	Прогнозований CPA	83,22 €
Прогнозований прибуток	4 156,1 €	Прогнозований прибуток	4 466 €	Прогнозований прибуток	4 775,9 €

Прогноз є приблизним і базується на припущенні, що ключові показники кампанії залишаються відносно стабільними. Результати реклами можуть змінюватися через аукціон, конкуренцію, сезонність, поведінку користувачів і зміни алгоритмів рекламної платформи.

Прогноз на основі лінійної регресії базується на наданих історичних даних і показує приблизний тренд, а не точний майбутній результат.

Рисунок 4.16 — Прогноз на основі лінійної регресії

Окремо перевірялася поведінка системи у випадку, коли даних для регресійного прогнозу недостатньо. Якщо користувач вводить лише одну кампанію вручну або завантажує CSV без щоденної динаміки, застосунок не повинен будувати регресійний прогноз. У такій ситуації система показує повідомлення про те, що для прогнозу на основі лінійної регресії потрібні історичні дані за днями або періодами.

Для прогнозу на основі лінійної регресії потрібні історичні дані за днями або періодами.

Рисунок 4.17 — Повідомлення про недостатність даних для регресійного прогнозу

Отже, тестування модуля прогнозування показало, що застосунок коректно формує майбутні сценарії на основі поточних показників кампанії, дозволяє змінювати рекламний бюджет і кількість днів прогнозу, а також не

виконує регресійний прогноз у випадках, коли історичних даних недостатньо. Це робить прогнозування більш обґрунтованим і зменшує ризик отримання ненадійних результатів.

4.6 Тестування візуалізації результатів

Наступним етапом було протестовано візуалізацію результатів аналізу. Метою цього тестування є перевірка того, чи коректно застосунок відображає розраховані показники у вигляді графіків і чи допомагають ці графіки користувачу швидше зрозуміти стан рекламної кампанії.

Після завантаження файлу `figure_4_4_sales_valid_daily.csv` та виконання аналізу система сформувала графіки на основі фактичних даних кампанії. Першим було перевірено графік порівняння фактичних значень із `benchmark`-показниками. Такий графік дозволяє побачити, які метрики кампанії є кращими за еталонні значення, а які відстають від них. Під час тестування було перевірено, що графік змінюється відповідно до обраного `benchmark`-режиму. Якщо обрано загальний `benchmark`, система порівнює показники кампанії із загальними еталонними значеннями. Якщо обрано режим без зовнішнього `benchmark`, застосунок не повинен показувати зовнішнє порівняння, щоб не вводити користувача в оману.

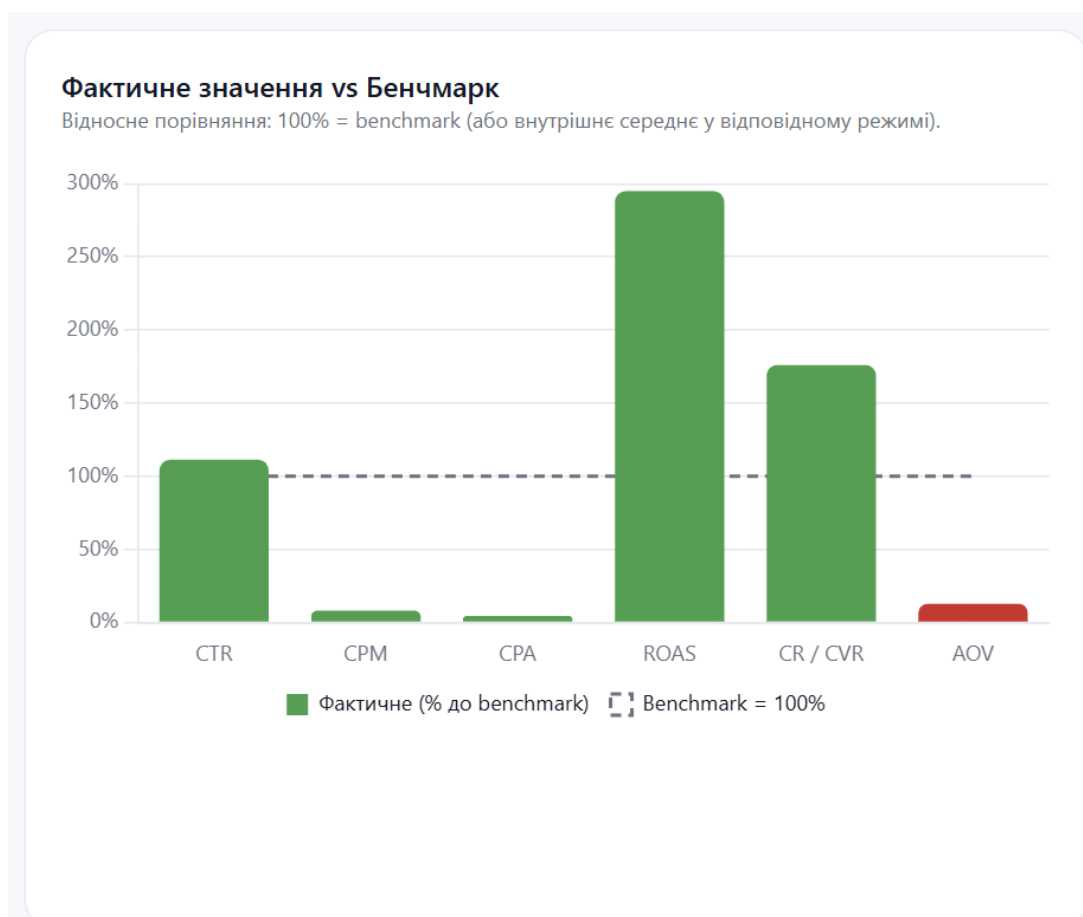


Рисунок 4.18 — Графік порівняння фактичних значень із benchmark

Окремо було перевірено графік щоденної динаміки. Оскільки файл `figure_4_4_sales_valid_daily.csv` містить колонку з датами, застосунок може побудувати зміну показників за днями. Такий графік дозволяє оцінити, чи були результати кампанії стабільними, чи певні дні суттєво відрізнялися від інших.

Під час тестування було підтверджено, що графік щоденної динаміки будується тільки тоді, коли у вхідному CSV-файлі є дата. Якщо дата відсутня, система не створює умовну часову динаміку, оскільки такий графік не мав би аналітичного змісту.

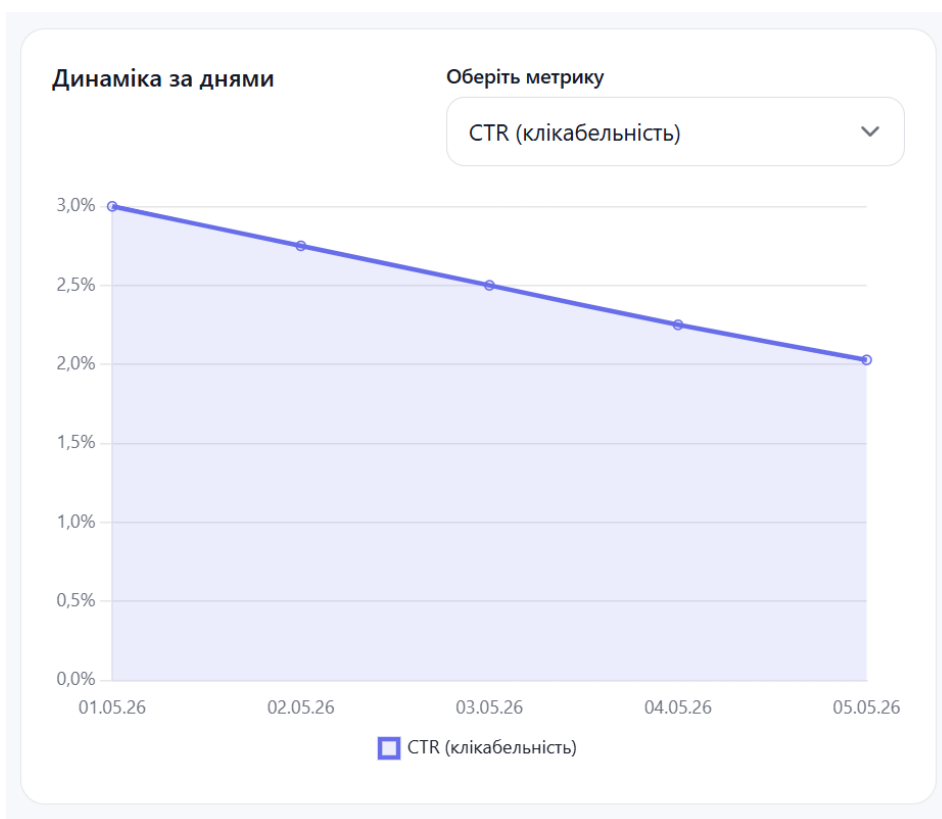


Рисунок 4.19 — Графік щоденної динаміки рекламної кампанії

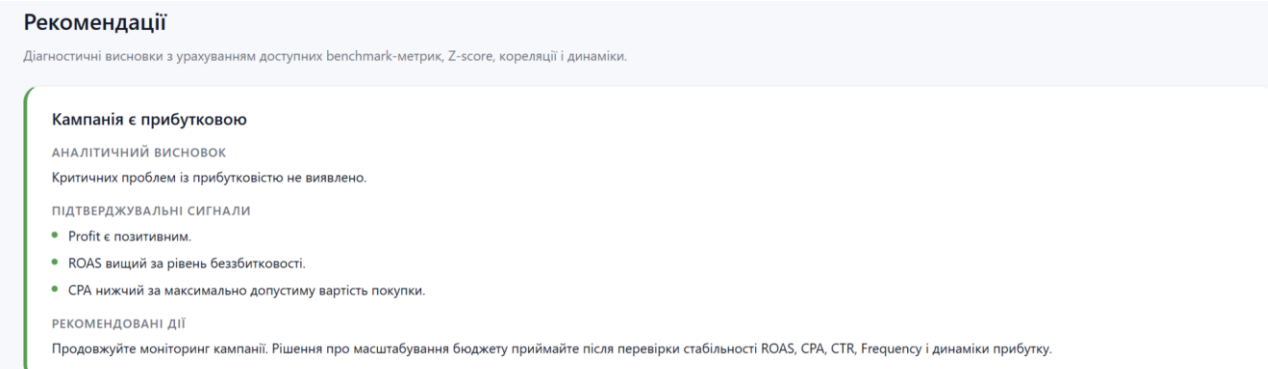
Також було протестовано візуалізацію в режимі порівняння кампаній. Для цього можна використати файл `04_sales_campaign_comparison.csv`. У цьому режимі графік повинен показувати різницю між кількома кампаніями за обраною метрикою. Наприклад, користувач може вибрати ROAS, CPA або прибуток і побачити, яка кампанія має кращий результат.

Крім змістовної правильності графіків, було перевірено зручність їх відображення в інтерфейсі. Графіки мають зрозумілі назви, підписи та не повинні перекривати інші елементи сторінки. Також перевірялося, що при зменшенні ширини вікна браузера блоки перебудовуються і залишаються читабельними.

4.7 Тестування блоку діагностичних рекомендацій

Окремим етапом було протестовано блок діагностичних рекомендацій. Його призначення полягає в тому, щоб не лише показати користувачу числові значення метрик, а й пояснити, які саме показники потребують уваги та які дії можуть бути доцільними для покращення результатів кампанії.

Для перевірки цього блоку використовувалися різні сценарії кампаній Продажі. Спочатку було протестовано кампанію з позитивними результатами на основі файлу `figure_4_4_sales_valid_daily.csv`. У цьому випадку кампанія має прибуток, достатньо високий ROAS і прийнятну вартість покупки. Тому система повинна сформулювати рекомендації, пов'язані не з критичними помилками, а з подальшою оптимізацією або можливістю масштабування кампанії.



Рекомендації
Діагностичні висновки з урахуванням доступних benchmark-метрик, Z-score, кореляції і динаміки.

Кампанія є прибутковою

АНАЛІТИЧНИЙ ВИСНОВОК
Критичних проблем із прибутковістю не виявлено.

ПІДТВЕРДЖУВАЛЬНІ СИГНАЛИ

- Profit є позитивним.
- ROAS вищий за рівень безбитковості.
- CPA нижчий за максимально допустиму вартість покупки.

РЕКОМЕНДОВАНІ ДІЇ
Продовжайте моніторинг кампанії. Рішення про масштабування бюджету приймайте після перевірки стабільності ROAS, CPA, CTR, Frequency і динаміки прибутку.

Рисунок 4.20 — Рекомендації для прибуткової кампанії

Далі було протестовано сценарій із проблемними показниками. Для цього можна використати кампанію з низьким доходом, високою вартістю покупки або збитковим результатом. У такому випадку система повинна показати попереджувальні або критичні рекомендації. Наприклад, якщо прибуток є від'ємним, рекомендація має звернути увагу користувача на те, що масштабування кампанії за поточних умов може збільшити збитки.

Приклад даних для проблемної кампанії Продажі:\

- Назва кампанії: Weak Sales Campaign
- Витрати на рекламу: 3000
- Дохід: 2500
- Покупки: 6
- Покази: 60000
- Кліки: 900
- Охоплення: 30000
- Середня валова маржа: 35

У цьому сценарії валовий прибуток становить: $2500 * 0,35 = 875$

Після врахування рекламних витрат чистий прибуток дорівнює: $875 - 3000 = -2125$

Отже, кампанія є збитковою, тому блок рекомендацій повинен показати користувачу критичне або попереджувальне повідомлення щодо низької економічної ефективності.

Рекомендації
Діагностичні висновки з урахуванням доступних benchmark-метрик.

Кампанія є збитковою

ОСНОВНА ПРОБЛЕМА
Кампанія не досягає рівня беззбитковості: фактичний ROAS нижчий за Break-even ROAS, а прибуток є негативним.

ПІДТВЕРДЖУВАЛЬНІ СИГНАЛИ

- Profit є негативним.
- ROAS нижчий за рівень беззбитковості.
- Actual CPA вищий за Max Profitable CPA.

РЕКОМЕНДОВАНІ ДІЇ
Не масштабуйте кампанію на цьому етапі. Спочатку потрібно низити CPA, покращити CR, підвищити AOV або збільшити маржу, щоб кампанія могла вийти на прибутковість.

Кампанія потребує комплексної оптимізації

ОСНОВНА ПРОБЛЕМА
Кампанія має проблеми на двох етапах: залучення кліків і конверсія після переходу.

ПІДТВЕРДЖУВАЛЬНІ СИГНАЛИ

- CTR нижчий за benchmark або внутрішню базу порівняння.
- CR / CVR нижчий за benchmark або внутрішню базу порівняння.

РЕКОМЕНДОВАНІ ДІЇ
Не масштабуйте кампанію на цьому етапі. Спочатку протестуйте нові креативи та рекламні повідомлення, потім перевірте релевантність посадкової сторінки, ціну, умови доставки та процес оформлення.

Рисунок 4.21 — Рекомендації для збиткової кампанії

Також було перевірено, що рекомендації формуються з урахуванням різних джерел аналітичних сигналів. Система аналізує не тільки одну метрику, а поєднує результати benchmark-порівняння, Campaign Score, статистичних відхилень, кореляційного аналізу та прогнозу. Завдяки цьому користувач отримує не набір випадкових порад, а структуровані висновки, пов'язані з фактичними результатами кампанії.

Під час тестування було звернено увагу на пріоритетність рекомендацій. Найважливіші проблеми мають відобразитися вище за другорядні. Наприклад, якщо кампанія є збитковою, це повідомлення повинно мати вищий пріоритет, ніж загальна порада щодо покращення CTR або тестування нових креативів. Такий підхід робить блок рекомендацій корисним для практичного прийняття рішень.

Отже, тестування блоку рекомендацій показало, що застосунок не обмежується простим відображенням числових метрик. Він інтерпретує результати аналізу та допомагає користувачу зрозуміти, які саме аспекти

рекламної кампанії потребують уваги. Це підвищує практичну цінність системи, оскільки користувач отримує не лише дані, а й аналітичні підказки для оптимізації реклами.

4.8 Аналіз результатів тестування та обмеження системи

За результатами тестування було підтверджено, що веб-застосунок коректно виконує основні сценарії роботи з рекламними кампаніями. Система приймає дані через ручну форму та CSV-файли, перевіряє їх перед аналізом, розраховує рекламні метрики, формує Campaign Score, будує прогноз, графіки та діагностичні рекомендації.

Під час тестування було перевірено як коректні, так і помилкові сценарії. У випадку правильних даних застосунок успішно виконував аналіз і відображав результати. У випадку некоректних даних, наприклад відсутнього доходу кампанії Продажі або більшої кількості переглядів посадкової сторінки, ніж кліків, система показувала повідомлення про помилку та не запускала розрахунки.

Також було підтверджено, що розраховані метрики збігаються з ручними контрольними розрахунками з урахуванням округлення. Статистичний аналіз, кореляція, прогнозування та графіки працюють тоді, коли для них є достатньо даних. Якщо даних недостатньо, застосунок повідомляє користувача про обмеження і не формує ненадійні висновки.

Разом із цим система має певні обмеження. Точність прогнозу залежить від якості та кількості вхідних даних. Benchmark-показники є орієнтовними, тому вони не можуть повністю врахувати специфіку кожного бізнесу, аудиторії

або рекламної платформи. Кореляційний аналіз показує лише статистичний зв'язок між метриками, але не доводить причинно-наслідкову залежність.

Крім того, застосунок працює у frontend-only форматі, тому не підключається автоматично до рекламних кабінетів. Користувач самостійно вводить дані або завантажує CSV-файли, тому якість аналізу безпосередньо залежить від правильності підготовлених даних.

Тестування показало, що розроблений веб-застосунок виконує поставлені функціональні вимоги та може використовуватися для аналізу ефективності рекламних кампаній. Він допомагає користувачу не лише отримати числові показники, а й інтерпретувати результати за допомогою порівняння, прогнозу, графіків і рекомендацій.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено веб-застосунок для аналізу та прогнозування ефективності рекламних кампаній. Його основна ідея полягає в тому, щоб користувач міг швидко оцінити результати реклами без складних таблиць і ручних розрахунків: ввести дані або завантажити CSV-файл, отримати ключові метрики, побачити проблемні місця та сформувані попередній прогноз.

У першому розділі було розглянуто предметну область цифрової реклами, основні типи рекламних кампаній та показники, які використовуються для оцінювання їх ефективності. Було визначено, що для повноцінного аналізу недостатньо дивитися лише на один показник, наприклад CTR або ROAS. Рекламну кампанію потрібно оцінювати комплексно: за вартістю, конверсією, прибутковістю, охопленням, залученням та порівнянням із benchmark-показниками.

У другому розділі було описано математичні методи, які використовуються в застосунку. До них належать розрахунок рекламних метрик, Z-score для пошуку відхилень, коефіцієнт кореляції Пірсона для виявлення зв'язків між показниками, прогнозування через рекламну воронку та лінійну регресію. Також було обґрунтовано використання інтегральної оцінки Campaign Score, яка дозволяє подати загальну ефективність кампанії у вигляді одного зрозумілого показника.

У третьому розділі було описано практичну реалізацію веб-застосунку. Було реалізовано ручне введення даних, завантаження CSV-файлів, перевірку коректності значень, розрахунок метрик, benchmark-порівняння, статистичний аналіз, прогнозування, графіки та блок рекомендацій. Застосунок працює у

frontend-only форматі, тому дані користувача не передаються на сервер, що є важливою перевагою з погляду приватності.

У четвертому розділі було проведено тестування розробленої системи. Перевірка показала, що застосунок коректно обробляє як правильні, так і помилкові дані. Якщо користувач вводить некоректні значення або завантажує неправильний CSV-файл, система повідомляє про помилку до початку розрахунків. Також було підтверджено, що основні рекламні метрики збігаються з ручними контрольними розрахунками, а прогноз, графіки, кореляційний аналіз і рекомендації працюють за наявності достатньої кількості даних.

Отже, поставлена мета дипломної роботи була досягнута. Розроблений веб-застосунок дозволяє аналізувати рекламні кампанії різних типів, оцінювати їх ефективність, знаходити слабкі місця та отримувати практичні рекомендації для оптимізації. Він може бути корисним для маркетологів, власників малого бізнесу та спеціалістів, яким потрібно швидко зрозуміти, наскільки результативно працює реклама.

Разом із цим система має простір для подальшого розвитку. У майбутньому її можна доповнити автоматичним імпортом даних із рекламних кабінетів, розширити набір benchmark-показників, додати більше галузей і реалізувати збереження історії аналізів. Проте вже в поточному вигляді застосунок виконує основні функції та може використовуватися як зручний інструмент для первинного аналізу ефективності рекламних кампаній.

ПЕРЕЛІК ПОСИЛАНЬ

1. Adsterra. Ad Metrics: 40+ KPIs for Marketers to Boost Ad Performance. URL: <https://adsterra.com/blog/ad-metrics> (дата звернення: 08.05.2026).
2. Triple Whale. Facebook Ad Benchmarks by Industry. URL: <https://www.triplewhale.com/blog/facebook-ads-benchmarks> (дата звернення: 22.04.2026).
3. WordStream by LocaliQ. Facebook Ads Benchmarks 2025. URL: <https://www.wordstream.com/blog/facebook-ads-benchmarks-2025> (дата звернення: 23.04.2026).
4. Meta Platforms, Inc. Meta Business Help Center. URL: <https://www.facebook.com/business/help> (дата звернення: 27.04.2026).
5. Статистичний аналіз результатів та оцінка z-score. URL: <https://atestor.ua/uk/labs/statystychnyy-analiz-rezultativ-ta-otsinka-z-score> (дата звернення: 09.05.2026).
6. Google LLC. Google Ads Help Center. URL: <https://support.google.com/google-ads> (дата звернення: 27.04.2026).
7. MDN Web Docs. JavaScript Reference. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference> (дата звернення: 03.05.2026).
8. Vite Contributors. Vite Guide. URL: <https://vite.dev/blog/cloudflare-supports-vite> (дата звернення: 06.05.2026).
9. PapaParse Contributors. PapaParse Documentation. URL: <https://www.papaparse.com/docs> (дата звернення: 06.05.2026).

ДОДАТОК А

Фрагменти програмного коду веб-застосунку

Розрахунок рекламних метрик:

```
function trafficMetrics(input) {
  const { adSpend, impressions, clicks } = input;
  const ctr = safeMul(safeDiv(clicks, impressions), 100);
  const cpc = safeDiv(adSpend, clicks);
  const cpm = safeMul(safeDiv(adSpend, impressions), 1000);
  return { ctr, cpc, cpm };
}

function salesMetrics(input) {
  const { adSpend, revenue, purchases } = input;
  const traffic = trafficMetrics(input);

  let grossProfit = null;
  let grossMargin = null;
  let profit = null;
  let breakEvenRoas = null;
  let maxProfitableCpa = null;

  if (input.profitMethod === PROFIT_METHOD.MARGIN) {
    const marginPct = input.averageGrossMargin;
    if (isFiniteNumber(marginPct) && marginPct > 0) {
      const marginRatio = marginPct / 100;
      grossMargin = marginRatio * 100;
      grossProfit = safeMul(revenue, marginRatio);
      profit = safeSub(grossProfit, adSpend);
      breakEvenRoas = safeDiv(1, marginRatio);
      maxProfitableCpa = safeDiv(grossProfit, purchases);
    }
  }

  const roas = safeDiv(revenue, adSpend);
  const cpa = safeDiv(adSpend, purchases);
  const cr = safeMul(safeDiv(purchases, input.clicks), 100);
  const aov = safeDiv(revenue, purchases);

  return {
    ...traffic,
    roas,
    grossProfit,
    grossMargin,
    profit,
    breakEvenRoas,
    cpa,
    cr,
    aov,
    maxProfitableCpa,
  };
};
```

```
}

```

Безпечні обчислення:

```
export function isFiniteNumber(value) {
  return typeof value === 'number' && Number.isFinite(value);
}

export function safeDiv(numerator, denominator) {
  if (!isFiniteNumber(numerator) || !isFiniteNumber(denominator))
return null;
  if (denominator === 0) return null;
  return numerator / denominator;
}

export function safeMul(...values) {
  let acc = 1;
  for (const v of values) {
    if (!isFiniteNumber(v)) return null;
    acc *= v;
  }
  return acc;
}

export function safeSub(a, b) {
  if (!isFiniteNumber(a) || !isFiniteNumber(b)) return null;
  return a - b;
}

```

Перевірка введених даних:

```
export function crossFieldErrors(campaignType, data) {
  const errors = [];
  const has = (id) => Number.isFinite(data[id]);

  if (has('clicks') && has('impressions') && data.clicks >
data.impressions) {
    errors.push(t('validation.clicksImpressions'));
  }

  if (has('reach') && has('impressions') && data.reach >
data.impressions) {
    errors.push(t('validation.reachImpressions'));
  }

  if (campaignType === CAMPAIGN_TYPE.SALES) {
    if (has('purchases') && has('clicks') && data.purchases >
data.clicks) {
      errors.push(t('validation.purchasesClicks'));
    }
  }
}

```

```

if (campaignType === CAMPAIGN_TYPE.LEADS) {
  if (has('leads') && has('clicks') && data.leads > data.clicks)
  {
    errors.push(t('validation.leadsClicks'));
  }

  if (has('qualifiedLeads') && has('leads') &&
    data.qualifiedLeads > data.leads) {
    errors.push(t('validation.qualifiedLeads'));
  }
}

if (campaignType === CAMPAIGN_TYPE.TRAFFIC) {
  if (has('landingPageViews') && has('clicks') &&
    data.landingPageViews > data.clicks) {
    errors.push(t('validation.lpvClicks'));
  }
}

return errors;
}

```

Статистичний аналіз:

```

export function mean(values) {
  const arr = cleanSingle(values);
  if (arr.length === 0) return null;

  let sum = 0;
  for (const v of arr) sum += v;

  return sum / arr.length;
}

export function stdDev(values) {
  const arr = cleanSingle(values);
  if (arr.length === 0) return null;

  const m = mean(arr);
  let acc = 0;

  for (const v of arr) {
    const d = v - m;
    acc += d * d;
  }

  return Math.sqrt(acc / arr.length);
}

export function zScore(value, dataset) {
  if (!isFiniteNumber(value)) return null;

```

```

const arr = cleanSingle(dataset);
if (arr.length < 2) return null;

const m = mean(arr);
const sd = stdDev(arr);

if (!isFiniteNumber(sd) || sd === 0) return null;

return (value - m) / sd;
}

export function pearson(xs, ys) {
  const pairs = cleanPairs(xs, ys);
  if (pairs.length < 3) return null;

  const mx = pairs.reduce((s, [x]) => s + x, 0) / pairs.length;
  const my = pairs.reduce((s, [, y]) => s + y, 0) / pairs.length;

  let cov = 0;
  let sx = 0;
  let sy = 0;

  for (const [x, y] of pairs) {
    const dx = x - mx;
    const dy = y - my;
    cov += dx * dy;
    sx += dx * dx;
    sy += dy * dy;
  }

  const denom = Math.sqrt(sx * sy);
  if (denom === 0) return null;

  return cov / denom;
}

```

Campaign Score:

```

function ratioScore(metricId, actual, target) {
  if (!isFiniteNumber(actual)) return null;
  if (!isFiniteNumber(target) || target === 0) return null;

  const m = meta(metricId);

  if (m.direction === 'higher_better') {
    return Math.min((actual / target) * 100, 100);
  }

  if (m.direction === 'lower_better') {
    return Math.min((target / actual) * 100, 100);
  }
}

```

```

    return null;
}

function applyBalanceCorrection(weightedScore, contributions,
weak, strong) {
  if (weak.length === 0 || strong.length === 0) return
weightedScore;

  const blend = weak.length === 1 ? 0.25 : weak.length === 2 ?
0.40 : 0.50;
  const weakAvg = weak.reduce((s, c) => s + c.score, 0) /
weak.length;

  return weightedScore * (1 - blend) + weakAvg * blend;
}

```

Відображення графіків:

```

export function ChartsBlock({ analysis, forecast }) {
  if (!analysis) {
    return StateBlock({
      kind: 'empty',
      title: t('state.empty'),
      hint: t('state.emptyHint'),
    });
  }

  const cards = [];

  if (hasActualVsBenchmarkData(analysis)) {
    cards.push(ActualVsBenchmarkChart({ analysis }).el);
  }

  if (analysis.task === ANALYSIS_TASK.CAMPAIGNS &&
    analysis.campaigns.length > 1) {
    cards.push(ByCampaignChart({ analysis }).el);
  }

  if (analysis.task === ANALYSIS_TASK.PERIODS &&
    analysis.campaigns.length > 1) {
    cards.push(ByPeriodChart({ analysis }).el);
  }

  const hasDaily = analysis.campaigns.some(
    (c) => c.daily && c.daily.length > 0
  );

  if (hasDaily) {
    cards.push(DailyChart({ analysis }).el);
  }

  if (forecast?.results) {

```

```
cards.push(...ScenarioCharts({
  campaignType: analysis.campaignType,
  scenarios: forecast.results,
  currency: analysis.currency,
}));
}

return el('div', { class: 'charts-grid' }, cards);
}
```