НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

До захисту допущено:
Завідувач кафедри
_____ Оксана ТИМОЩУК
«___»_____2026 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 Системний аналіз
на тему: «Система підтримки прийняття рішень для аналізу експериментальних даних у цифрових продуктах»

Виконав:
Студент IV курсу, групи КА-22
Черток Микола

_____

Керівник:
Ст. викладач кафедри ММСА, доктор філософії
Левенчук Людмила Борисівна

_____

Консультант з економічного розділу:
Доцент кафедри ТПЕ, к.е.н.
Рощина Надія Василівна

_____

Консультант з нормоконтролю:
Асистент кафедри ММСА,доктор філософії
Канцедал Георгій Олегович

_____

Рецензент:
Доцент кафедри ШІ, доктор філософії
доцент Гуськова Віра Геннадіївна

_____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 124 «Системний аналіз»
Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Оксана ТИМОЩУК
«___»_____2026 р.

ЗАВДАННЯ
на дипломну роботу студенту/ці
ПІП

1. Тема роботи “Система підтримки прийняття рішень для аналізу експериментальних даних у цифрових продуктах”, керівник роботи Ст. викладач кафедри ММСА, доктор філософії Людмила Левенчук, затверджені наказом по університету 27.05.2026 р. № 1944-с
2. Термін подання студентом роботи «10» червня 2026 р.
3. Вихідні дані до роботи: науково-методична література за темою роботи; відкриті набори даних A/B-тестів цифрових продуктів; платформи Optimizely, GrowthBook та внутрішні системи технологічних компаній як орієнтири порівняння; програмні засоби Python (pandas, scipy, numpy, streamlit),матеріали переддипломної практики.
4. Зміст роботи: дослідження предметної області та постановка проблеми; аналіз методів статистичного оцінювання результатів A/B-тестів (z-тест, t-тест Велча, bootstrap, байєсівський підхід); розробка математичної моделі формування

аналітичного репорту; проектування та програмна реалізація системи підтримки прийняття рішень як веб-застосунку; оцінка ефективності та економічний аналіз розробленого програмного продукту.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): схема класифікації метрик A/B-тестування та типових помилок інтерпретації; архітектурна схема системи підтримки прийняття рішень (DSS); схема процедури bootstrap для оцінки довірчого інтервалу; порівняльна таблиця існуючих платформ; скріншоти інтерфейсу системи (дашборд, health-check, сегментний аналіз); презентація.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Функціонально-вартісний аналіз	Рощина Н.В., доц., к.е.н.		

7. Дата видачі завдання 27.05.2026

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми бакалаврської дипломної роботи (БДР), опанування ДСТУ 3008:2015	26.02.2026-28.02.2026	виконано
2	Огляд теоретичної інформації по темі роботи	20.04.2026-24.04.2026	виконано
3	Проектування і розробка програмного забезпечення.	04.05.2026-08.05.2026	виконано
4	Оформлення пояснювальної записки (ПЗ), перевірка правильності структури та оформлення	11.05.2026-15.05.2026	виконано
5	Аналіз результатів	18.05.2026-22.05.2026	виконано
6	Остаточне оформлення ПЗ, підготовка презентації	23.05.2025-10.06.2025	виконано
7	Передзахист БДР	06.06.2025-10.06.2025	виконано

Студент/ка _____ Микола ЧЕРТОК

Керівник _____ Людмила ЛЕВЕНЧУК

РЕФЕРАТ

Дипломна робота: 134 с., 15 табл., 8 рис., 1 додаток, 19 джерел.

A/B ТЕСТУВАННЯ, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, СТАТИСТИЧНИЙ АНАЛІЗ, BOOTSTRAP, SAMPLE RATIO MISMATCH, ДОВІРЧИЙ ІНТЕРВАЛ

Об'єктом дослідження у даній роботі є процеси аналізу та інтерпретації результатів контрольованих онлайн-експериментів (A/B-тестів) у цифрових продуктах — мобільних застосунках, SaaS-сервісах та вебплатформах. Предметом дослідження є математичні моделі та програмні методи статистичного оцінювання ефектів, діагностики якості експериментів і підтримки прийняття рішень на основі структурованої системи первинних, вторинних та охоронних метрик.

Метою дипломної роботи є підвищення обґрунтованості та якості продуктових рішень за результатами A/B-тестування шляхом проектування, програмної реалізації та дослідження системи підтримки прийняття рішень, яка автоматизує повний аналітичний цикл — від завантаження та валідації сирих експериментальних даних до формування структурованого аналітичного репорту з виявленням конфліктних сигналів між метриками.

У межах практичної реалізації проєкту розроблено веб-застосунок мовою Python на основі фреймворку Streamlit. Реалізовано модуль завантаження та автоматизованої валідації CSV-файлів з даними A/B-тестів, статистичний модуль (z-тест для пропорцій, t-тест Велча, байєсівська оцінка через Beta-розподіл), модуль bootstrap-аналізу для побудови довірчих інтервалів метрик з асиметричним розподілом, health-check модуль з автоматизованою перевіркою Sample Ratio Mismatch та балансу сегментів, а також модуль сегментованого аналізу гетерогенності ефекту за user properties. Ключовим практичним результатом є формування єдиного структурованого аналітичного

репорту, що агрегує сигнали від primary, secondary та guardrail метрик і містить блок автоматичних попереджень при виявленні аномалій у даних або суперечливих результатів між метриками.

Перспективи подальшого розвитку дослідження полягають у розширенні системи шляхом інтеграції методів послідовного аналізу (Sequential Testing) для коректної обробки результатів при достроковому завершенні тестів, а також у реалізації підтримки багатоваріантних тестів (MVT) та автоматизованого розрахунку мінімального розміру вибірки на етапі планування експерименту.

ABSTRACT

Thesis: 134 p., 15 tab., 8 Fig., 1 appendix, 19 sources.

A/B TESTING, DECISION SUPPORT SYSTEM, STATISTICAL ANALYSIS, BOOTSTRAP, SAMPLE RATIO MISMATCH, CONFIDENCE INTERVAL

The object of research in this work is the processes of analysis and interpretation of results of controlled online experiments (A/B tests) in digital products — mobile applications, SaaS services, and web platforms. The subject of the study is mathematical models and software methods for statistical estimation of effects, experiment quality diagnostics, and decision support based on a structured system of primary, secondary, and guardrail metrics.

The purpose of the thesis is to improve the validity and quality of product decisions based on A/B testing results by designing, implementing, and investigating a decision support system that automates the complete analytical cycle — from loading and validating raw experimental data to generating a structured analytical report with identification of conflicting signals between metrics.

As part of the practical implementation, a web application was developed in Python using the Streamlit framework. The system includes a module for loading and automated validation of CSV files with A/B test data, a statistical module (z-test for proportions, Welch's t-test, Bayesian estimation via Beta distribution), a bootstrap analysis module for building confidence intervals for metrics with asymmetric distributions, a health-check module with automated Sample Ratio Mismatch detection and segment balance verification, and a segmented analysis module for detecting heterogeneity of treatment effects across user properties. The key practical result is a unified structured analytical report that aggregates signals from primary, secondary, and guardrail metrics and includes an automated warnings block triggered by data anomalies or conflicting metric results.

Prospects for further development include extending the system by integrating Sequential Testing methods for correct handling of results under early stopping conditions, as well as implementing support for multivariate tests (MVT) and automated minimum sample size calculation at the experiment planning stage.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасних умовах розвитку цифрової економіки продуктові рішення дедалі рідше ґрунтуються на інтуїції чи експертних оцінках — натомість технологічні компанії спираються на дані контрольованих онлайн-експериментів. A/B-тестування як метод рандомізованого порівняльного аналізу стало стандартом у розробці мобільних застосунків, SaaS-сервісів та вебплатформ: за оцінками галузевих досліджень, провідні технологічні компанії щороку проводять тисячі паралельних експериментів, а рішення про зміну ключових функцій продукту без підтвердження тестом вважається методологічно неприйнятним. Разом із зростанням культури експериментування загострюється і проблема коректності інтерпретації результатів: некоректне розуміння p-value, передчасне завершення тестів, ігнорування структурних вад у даних та прихована неоднорідність ефекту по сегментах аудиторії систематично призводять до хибних висновків і продуктових рішень, що не відтворюються в реальних умовах [1].

Незважаючи на існування низки комерційних та відкритих платформ для A/B-тестування, жодна з них не забезпечує комплексного вирішення зазначених проблем в єдиному аналітичному середовищі, доступному для широкого кола продуктових команд без значних інженерних ресурсів. Комерційні рішення обмежені у статистичних методах і не підтримують непараметричних підходів та автоматизованої діагностики якості тесту. Відкриті платформи потребують складної інфраструктури для розгортання. Внутрішні системи технологічних лідерів, попри функціональну повноту, є недоступними для зовнішнього використання. Це формує незакритий запит на спеціалізований інструмент, здатний автоматизувати повний аналітичний цикл — від валідації сирих даних до формування структурованого репорту з виявленням конфліктних сигналів між метриками.

Мета дослідження — розробити та реалізувати систему підтримки прийняття рішень для аналізу результатів A/B-тестування у цифрових продуктах, яка автоматизує статистичне оцінювання ефектів, діагностику якості експерименту та формування структурованого аналітичного репорту на основі системи первинних, вторинних та охоронних метрик.

Відповідно до мети були поставлені такі завдання: дослідити предметну область A/B-тестування у цифрових продуктах, систематизувати типові проблеми інтерпретаці…Відповідно до мети були поставлені такі завдання: дослідити предметну область A/B-тестування у цифрових продуктах, систематизувати типові проблеми інтерпретації результатів та проаналізувати існуючі платформи і їхні обмеження; формалізувати математичний апарат статистичного аналізу — z-тест для пропорцій, t-тест Велча, bootstrap-оцінювання довірчих інтервалів, байєсівський підхід через Beta-розподіл — та обґрунтувати методологію health-check діагностики якості тесту; спроектувати та реалізувати архітектуру системи підтримки прийняття рішень як інтерактивного веб-застосунку з модулями валідації даних, статистичного аналізу, bootstrap, health-check, сегментованого аналізу та формування репорту; провести техніко-економічний аналіз розробленого програмного продукту та оцінити доцільність його застосування у продуктових командах.

Об'єкт дослідження — процеси аналізу та інтерпретації результатів контрольованих онлайн-експериментів у цифрових продуктах.

Предмет дослідження — математичні моделі та програмні методи статистичного оцінювання ефектів A/B-тестів, діагностики якості експериментів і підтримки прийняття рішень на основі структурованої системи метрик.

Наукова новизна полягає у розробці методології формування єдиного структурованого аналітичного репорту, що інтегрує результати параметричного та непараметричного (bootstrap) статистичного аналізу, автоматизованої

health-check діагностики Sample Ratio Mismatch і балансу сегментів та сегментованого аналізу гетерогенності ефекту в межах єдиної архітектури системи підтримки прийняття рішень, орієнтованої на продуктові команди цифрових продуктів.

Практична значущість роботи полягає у тому, що розроблений веб-застосунок може бути безпосередньо застосований продуктовими аналітиками та менеджерами для аналізу результатів A/B-тестів без необхідності написання коду або розгортання інженерної інфраструктури. Система скорочує час від завершення експерименту до прийняття обґрунтованого продуктового рішення та знижує ймовірність систематичних аналітичних помилок завдяки автоматизованій діагностиці та структурованій подачі результатів.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ПРОБЛЕМИ

1.1 Сутність та принципи A/B тестування

Розвиток цифрових продуктів нерозривно пов'язаний з необхідністю постійного вдосконалення користувацького досвіду, підвищення ефективності бізнес-процесів та збільшення ключових показників залучення й монетизації аудиторії. Однак будь-яке продуктове рішення несе в собі ризик: зміна, що здається очевидно корисною на етапі гіпотези, може виявитися нейтральною або навіть шкідливою в умовах реальної взаємодії користувачів із системою. Саме тому сучасна продуктова розробка спирається на методологію контрольованих онлайн-експериментів, центральне місце в якій посідає A/B тестування.

A/B тестування являє собою метод порівняльного аналізу двох варіантів продукту або його окремого компонента шляхом одночасного представлення кожного з них різним, але репрезентативним групам реальних користувачів. Варіант A, який прийнято називати контрольним, відображає поточний стан системи, тоді як варіант B — тестовий — містить запропоновану зміну. За результатами спостереження за поведінкою обох груп протягом визначеного часового проміжку приймається рішення про доцільність впровадження тестової зміни у продукт. Принципова особливість методу полягає в тому, що рішення ґрунтується не на суб'єктивних оцінках або аналітичних припущеннях, а на емпіричних даних, отриманих безпосередньо від цільової аудиторії продукту.

Концептуальну основу A/B тестування становить методологія рандомізованого контрольованого експерименту (Randomized Controlled Trial, RCT), яка широко застосовується у клінічних дослідженнях, соціальних науках та економіці. Перенесення цієї методології у середовище цифрових продуктів відбулося на початку 2000-х років, коли такі компанії, як Microsoft, Google та

Amazon, почали систематично використовувати онлайн-експерименти для оптимізації своїх сервісів. Ключовою умовою коректного застосування RCT є рандомізація — процедура випадкового та незалежного призначення кожного користувача до однієї з груп, яка забезпечує статистичну еквівалентність груп за всіма ненаблюдуваними характеристиками та усуває систематичні зміщення у вимірах ефекту.

Механізм проведення A/B тесту в цифровому продукті, зокрема в мобільному застосунку або SaaS-сервісі, передбачає декілька ключових етапів. На першому етапі формулюється дослідницька гіпотеза: визначається конкретна зміна у продукті, яку планується протестувати, та очікуваний напрям її впливу на обрану метрику. Прикладом може служити гіпотеза про те, що спрощення форми реєстрації скоротить кількість кроків та підвищить конверсію нових користувачів у вебсервісі. На другому етапі визначаються одиниця рандомізації та параметри розподілу трафіку. Найпоширенішою одиницею рандомізації є ідентифікатор користувача, що забезпечує стабільність варіанта для кожного конкретного користувача протягом усього часу проведення тесту — властивість, відома як «sticky assignment».

Математичну основу A/B тестування утворює апарат статистичного тестування гіпотез. Перед початком експерименту формулюються нульова та альтернативна гіпотези. Нульова гіпотеза H_0 стверджує, що між контрольною та тестовою групами не існує статистично значущої різниці у значенні досліджуваної метрики, тобто спостережувана різниця зумовлена виключно випадковими коливаннями. Альтернативна гіпотеза H_1, навпаки, стверджує, що тестовий варіант справляє реальний ефект на вимірювану метрику. Задача статистичного аналізу полягає у визначенні, чи є зібраних даних достатньо для відхилення нульової гіпотези з прийнятним рівнем статистичної впевненості.

Важливою характеристикою коректно спланованого A/B тесту є дотримання принципу статистичної ізоляції між групами. Він полягає в тому, що

поведінка користувача, який потрапив до однієї групи, не повинна впливати на поведінку користувачів іншої групи. Це особливо актуально для соціальних функцій мобільних застосунків та вірусних механік SaaS-продуктів, де порушення ізоляції може суттєво викривити результати виміру ефекту. Окрім ізоляції, необхідно забезпечити репрезентативність обох груп, тобто відповідність їхнього складу загальній структурі аудиторії продукту за ключовими характеристиками — платформою, географією, часом реєстрації та іншими значущими параметрами.

Методологічно A/B тестування необхідно відрізняти від споріднених методів онлайн-експериментів. На відміну від багатоваріантного тестування (Multivariate Testing, MVT), яке дозволяє одночасно перевіряти декілька змін та досліджувати їх взаємодію між собою, класичний A/B тест зосереджений на ізольованому дослідженні одного чітко визначеного чинника. Це спрощує інтерпретацію результатів, однак вимагає проведення серії послідовних тестів за наявності кількох конкуруючих гіпотез. A/A тестування, при якому обидві групи отримують ідентичний варіант продукту, застосовується для верифікації коректності самої інфраструктури розподілу трафіку та виявлення системних зміщень ще до початку реальних експериментів.

Окрему увагу необхідно приділити питанню визначення достатнього розміру вибірки для забезпечення статистичної потужності тесту. Мінімальна кількість користувачів у кожній групі визначається, виходячи з очікуваного розміру ефекту (мінімальної значущої різниці, яку необхідно виявити), рівня значущості α та бажаної потужності тесту $1-\beta$. Ці параметри відображають баланс між ризиком хибнопозитивного результату (помилка першого роду) та ризиком пропустити реальний ефект (помилка другого роду). У практиці цифрових продуктів стандартним є рівень значущості $\alpha = 0{,}05$ та потужність тесту $1-\beta = 0{,}80$, що відповідає 80% ймовірності виявлення ефекту за умови його реального існування. Тривалість тесту також має практичне значення:

занадто короткий тест ризикує не охопити природні коливання в поведінці аудиторії, пов'язані зі щотижневою або місячною сезонністю [2].

Практичне застосування A/B тестування у цифрових продуктах охоплює надзвичайно широке коло задач. У мобільних застосунках метод використовується для оптимізації онбордингових потоків, дослідження ефективності push-сповіщень, тестування нових функцій та оцінки альтернативних моделей монетизації. У SaaS-продуктах A/B тести застосовуються для підвищення конверсії у платні підписки, оптимізації сторінок ціноутворення та вдосконалення інтерфейсів налаштування. У веб сервісах метод дозволяє оцінювати зміни в алгоритмах рекомендацій, пошукової видачі або персоналізованого контенту. В усіх цих випадках A/B тестування забезпечує емпіричне підґрунтя для прийняття продуктових рішень та суттєво знижує ризик впровадження змін, що мають негативний вплив на бізнес-показники.

Слід зазначити, що попри концептуальну простоту, практична реалізація A/B тестування на промисловому масштабі є складним інженерним і аналітичним завданням [1]. Вона потребує надійної інфраструктури розподілу трафіку, систем збору та зберігання подієвих даних, інструментів автоматизованого статистичного аналізу та механізмів інтерпретації результатів, доступних для широкого кола продуктових спеціалістів. Саме сукупність цих вимог формує запит на спеціалізовані системи підтримки прийняття рішень, орієнтовані на аналіз результатів A/B тестів, розгляду яких присвячена дана робота. Визначення ключових метрик, які стають об'єктом виміру в таких системах, є предметом наступного підрозділу.

1.2 Основні метрики в продуктовій аналітиці

Ефективність A/B тестування як інструменту підтримки продуктових рішень безпосередньо залежить від якості обраних метрик. Метрика — це кількісний показник, що описує певний аспект поведінки користувача або стану продукту і слугує операціональним виміром гіпотези, покладеної в основу експерименту. Вибір метрики є концептуальним рішенням, яке передує будь-яким статистичним розрахункам: некоректно обрана метрика здатна призвести до хибних висновків навіть за бездоганно проведеного тесту. У продуктовій аналітиці цифрових продуктів метрики прийнято класифікувати за їхньою роллю у процесі прийняття рішень на три категорії: первинні (primary), вторинні (secondary) та охоронні (guardrail).

Первинна метрика є головним критерієм оцінки успішності конкретного A/B тесту. Вона безпосередньо відображає ту характеристику продукту, на поліпшення якої спрямована тестована зміна, і саме за її динамікою ухвалюється рішення про впровадження або відхилення варіанта B. У кожного окремого тесту має бути лише одна первинна метрика — це запобігає проблемі множинних порівнянь і забезпечує однозначність інтерпретації. Вторинні метрики доповнюють аналіз: вони не є критеріями рішення самі по собі, але дозволяють глибше зрозуміти механізм виявленого ефекту та перевірити, чи відповідає поведінка суміжних показників очікуванням. Наприклад, якщо первинна метрика тесту — конверсія у платну підписку SaaS-продукту, вторинними можуть виступати кількість сесій до конверсії та глибина використання функціонального пробного доступу.

Охоронні метрики (guardrail metrics) відіграють особливу роль у системі аналізу A/B тестів: вони призначені не для вимірювання очікуваного ефекту, а для виявлення його небажаних побічних наслідків. Логіка їх застосування ґрунтується на тому, що оптимізація продукту за однією метрикою нерідко

супроводжується деградацією інших показників, що не відстежуються явно. Так, агресивна оптимізація конверсії у мобільному застосунку через спрощення реєстраційного потоку може збільшити частку неякісних реєстрацій та знизити Retention на 7-й день. Зростання revenue per session через збільшення кількості рекламних блоків може призвести до зниження часу сесії та зростання показника відмов. Введення guardrail-метрик до аналізу дозволяє захистити продукт від таких компромісів і гарантує, що впроваджена зміна не погіршує користувацький досвід або здоров'я бізнес-показників за межами вимірюваного ефекту. Класифікацію метрик за типами наведено у таблиці 1.1.

Таблиця 1.1 – Класифікація метрик за роллю в A/B тестуванні

Тип метрики	Призначення	Приклади	Роль у A/B тесті
Primary (первинна)	Головний критерій успіху	Conversion rate, LTV,ARPPU	Визначає рішення про впровадження
Secondary (вторинна)	Додаткова валідація ефекту	Session duration, Retention D7	Підтверджує або уточнює primary
Guardrail (охоронна)	Захист від небажаних ефектів	Page load time, Churn rate, Cancel Rate D3	Блокує впровадження при деградації

Серед конкретних метрик продуктової аналітики центральне місце займає показник конверсії (Conversion Rate, CVR , CR). У загальному вигляді conversion rate визначається як відношення кількості користувачів, які виконали цільову дію, до загальної кількості користувачів, що потрапили до відповідної когорти. Цільовою дією може виступати реєстрація, активація, здійснення першої покупки, перехід на платний план або будь-яка інша подія, що відображає просування користувача вздовж воронки продукту. Формально CVR для групи g визначається як:

$$CVR_g = \frac{N_{conv,g}}{N_{total,g}}, \tag{1.1}$$

де $N_{conv,g}$ — кількість користувачів групи g, які виконали цільову дію;

$N_{total,g}$ — загальна кількість користувачів групи g, що брали участь у тесті.

CVR є бінарною метрикою, оскільки кожен окремий користувач або виконав цільову дію (значення 1), або ні (значення 0). Ця властивість визначає характер статистичного тесту, який застосовується для порівняння груп.

Показник uplift описує відносну зміну метрики між тестовим та контрольним варіантами і є основним засобом вираження ефекту A/B тесту у продуктовій аналітиці. На відміну від абсолютної різниці між значеннями метрики, відносний uplift нормалізований відносно базового рівня контрольної групи, що робить результати порівнянними між тестами з різними базовими рівнями метрики. Відносний uplift розраховується за формулою:

$$Uplift = (CVR_b - CVR_a)/CVR_a \times 100\%, \quad (1.2)$$

де CVR_b — значення метрики у тестовій групі (варіант B);

CVR_a — значення метрики у контрольній групі (варіант A).

Позитивне значення uplift свідчить про поліпшення метрики у тестовому варіанті, від'ємне — про її погіршення. Наприклад, якщо conversion rate у контрольній групі мобільного застосунку становив 4,2%, а у тестовій — 4,7%, відносний uplift дорівнює приблизно 11,9%, що в залежності від обсягу аудиторії може мати суттєве практичне значення для бізнесу.

Поряд із конверсією та uplift у продуктовій аналітиці широко застосовується клас безперервних метрик. До них належать середній дохід на користувача (Average Revenue Per User, ARPU), середня вартість замовлення (Average Order Value, AOV), тривалість сесії та частота повернення. Безперервні метрики, на відміну від бінарних, описують інтенсивність певного аспекту поведінки, а не сам факт його наявності. Вони характеризуються, як правило, суттєво правосторонньою асиметрією розподілу: невелика частка користувачів генерує непропорційно велику частку доходу або активності. Ця особливість має

важливі наслідки для вибору статистичного методу — зокрема, саме через неї для аналізу безперервних метрик у A/B тестах доцільно застосовувати непараметричні методи, зокрема bootstrap, що детальніше розглядається у другому розділі роботи.

Окремої уваги заслуговує концепція метрики північної зірки (North Star Metric, NSM) та ієрархії метрик у продуктовій аналітиці. North Star Metric — це єдиний узагальнений показник, який найбільш повно відображає цінність продукту для користувача і при цьому корелює з довгостроковим зростанням бізнесу. Для стрімінгового сервісу такою метрикою може бути кількість переглянутих годин на активного користувача на тиждень, для SaaS-продукту — кількість активних команд, що використовують ключовий функціонал. Метрики нижчих рівнів ієрархії — зокрема, ті, що використовуються у конкретних A/B тестах — мають бути логічно пов'язані з NSM і відображати конкретні чинники, що визначають її зміну. Така ієрархічна структура забезпечує узгодженість між короткостроковими результатами окремих тестів і довгостроковими цілями розвитку продукту.

Принциповою проблемою, що виникає при роботі з кількома метриками одночасно, є необхідність узгодження суперечливих сигналів. У реальних A/B тестах цифрових продуктів нерідко трапляється ситуація, коли primary метрика демонструє позитивний ефект, а одна або кілька guardrail-метрик при цьому погіршуються. Наприклад, агресивне скорочення реєстраційної форми у мобільному застосунку може підвищити конверсію у реєстрацію, але водночас знизити якість заповнення профілю та середнє утримання на 30-й день. Саме для вирішення таких неоднозначних ситуацій необхідна система підтримки прийняття рішень, здатна агрегувати сигнали від різних метрик і формувати зважену рекомендацію з урахуванням їхньої пріоритетності. Формалізація цього механізму є одним із центральних завдань даної роботи і розглядається у підрозділі 2.5.

Таким чином, система метрик є не просто набором показників для вимірювання, а структурованою концептуальною основою, що визначає логіку інтерпретації результатів A/B тесту. Коректна класифікація метрик на primary, secondary та guardrail, розуміння природи бінарних і безперервних показників, а також усвідомлення їхньої ролі у загальній ієрархії продуктових цілей є необхідними передумовами для побудови надійної системи аналізу експериментів. Питання про те, які платформи та інструменти існують для підтримки цих процесів у промислових умовах, розглядається у наступному підрозділі.

1.3 Огляд сучасних платформ для A/B тестування

Зростання масштабу онлайн-експериментів у технологічних компаніях сформувало стійкий попит на спеціалізовані платформи, здатні автоматизувати ключові етапи проведення A/B тестів — від розподілу трафіку до статистичного аналізу результатів. Сучасний ринок таких рішень є неоднорідним і охоплює комерційні SaaS-платформи, відкриті бібліотеки та внутрішні корпоративні системи великих технологічних компаній. Кожна з категорій має характерні переваги та обмеження, що визначають придатність конкретного рішення для тих чи інших умов застосування.

Серед комерційних платформ найбільшого поширення набули Optimizely та VWO (Visual Website Optimizer). Платформа Optimizely орієнтована переважно на великі підприємства та пропонує широкий набір функцій для проведення веб і мобільних експериментів, включаючи візуальний редактор змін, управління аудиторіями та базові інструменти статистичного аналізу. Однак статистичний модуль платформи ґрунтується виключно на частотному підході з фіксованим горизонтом, без підтримки непараметричних методів оцінки, зокрема bootstrap. VWO пропонує схожий функціонал із дещо нижчим

ціновим порогом, проте також не забезпечує гнучкого управління guardrail-метриками та не підтримує автоматизований health-check аналіз рівномірності розподілу трафіку. Обидві платформи є закритими системами, що унеможливлює розширення їхнього статистичного ядра силами команди аналітики.

У сегменті відкритих рішень привертає увагу платформа GrowthBook, яка дозволяє командам розгортати власну інфраструктуру A/B тестування з відкритим вихідним кодом. GrowthBook підтримує визначення guardrail-метрик, базову сегментацію та інтеграцію з різними джерелами даних. Проте для повноцінного функціонування платформа потребує значної інженерної інфраструктури — окремих сервісів збору подій, сховищ даних та налаштованих конвеєрів обробки — що робить її малодоступною для невеликих продуктових команд без виділеного data engineering ресурсу. Бібліотека PlanOut, розроблена в компанії Meta, вирішує суто інженерну задачу рандомізованого призначення користувачів до варіантів експерименту і не включає жодного аналітичного функціоналу: статистичний аналіз результатів залишається повністю поза її сферою.

Якісно відмінний рівень можливостей демонструють внутрішні платформи великих технологічних компаній. Система Experimentation Platform компанії Airbnb, детально описана у публічних технічних матеріалах, включає повний цикл аналізу: автоматизовану перевірку коректності розподілу трафіку (SRM-перевірку), bootstrap-оцінку довірчих інтервалів для асиметричних метрик доходу, сегментний аналіз гетерогенності ефекту та систему автоматичних попереджень при порушенні guardrail-показників. Аналогічну архітектуру має платформа XP компанії Netflix, яка орієнтована на роботу з великими обсягами даних і підтримує байєсівські методи оцінки поряд із частотними. Спільною рисою цих систем є те, що вони розроблялися протягом кількох років силами

великих команд і є невід'ємною частиною внутрішньої інфраструктури відповідних компаній, а отже, недоступні для зовнішнього використання [3].

Порівняльний аналіз розглянутих платформ за ключовими аналітичними характеристиками подано у таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз платформ для A/B тестування

Платформа	Тип	Bootstrap	Сегментація	Guardrail	Обмеження
Optimizely	Комерційна SaaS	Ні	Так	Обмежено	Висока вартість, закрита архітектура
VWO	Комерційна SaaS	Ні	Так	Ні	Обмежені статистичні методи
GrowthBook	Open-source	Ні	Так	Так	Потребує інфраструктури
PlanOut (Meta)	Open-source бібліотека	Ні	Ні	Ні	Лише розподіл трафіку, без аналізу
Внутрішні системи (Airbnb, Netflix)	Корпоративна	Так	Так	Так	Недоступні публічно

Аналіз представлених рішень виявляє характерний розрив між доступними комерційними інструментами та функціональністю, необхідною для повноцінного аналізу результатів A/B тестів у цифрових продуктах. Комерційні платформи, попри зручний інтерфейс, обмежені у статистичних методах і не підтримують ані bootstrap-аналізу, ані автоматизованих health-check перевірок. Відкриті рішення потребують значних інженерних ресурсів для розгортання. Внутрішні системи технологічних лідерів, попри свою функціональну повноту, є недоступними для широкого загалу продуктових команд [4].

Окремо слід відзначити загальну тенденцію, що спостерігається у галузі: зростання продуктових команд та обсягу проведених експериментів формує

запит на спеціалізовані аналітичні інструменти, здатні агрегувати результати по кількох метриках, проводити автоматизовану діагностику якості тесту та формувати зрозумілі рекомендації для осіб, що приймають рішення. Цей запит поки що не задоволений жодним з доступних на ринку рішень у повному обсязі, що підтверджує актуальність розробки спеціалізованої системи підтримки прийняття рішень, описаної у даній роботі. Конкретні проблеми, що виникають у процесі інтерпретації результатів A/B тестів і визначають вимоги до такої системи, розглядаються у наступному підрозділі.

1.4 Проблеми інтерпретації результатів A/B тестів

Навіть за умови технічно коректного проведення A/B тесту — з належною рандомізацією, достатнім розміром вибірки та дотриманням ізоляції між групами — процес інтерпретації його результатів залишається джерелом численних аналітичних помилок. Ці помилки мають різну природу: одні пов'язані з порушенням статистичних припущень, інші — з некоректним застосуванням статистичних критеріїв, треті — зі структурними вадами самого експерименту, що залишаються непоміченими без спеціалізованої діагностики. Розуміння типових проблем інтерпретації є необхідною передумовою для проектування системи аналізу, що здатна їх виявляти та враховувати.

Першою і найпоширенішою проблемою є некоректна інтерпретація p-value. У продуктовій практиці p-value нерідко сприймається як «ймовірність того, що гіпотеза вірна» або «ймовірність того, що результат є випадковим», тоді як формально він означає ймовірність отримати спостережувану або більш екстремальну різницю між групами за умови, що нульова гіпотеза є істинною. Ця тонка, але принципова відмінність призводить до систематичного завищення впевненості у висновках. Зокрема, досягнення порогового значення $p < 0,05$ не означає, що ефект є практично значущим або що він обов'язково відтвориться у

наступному тесті. Ігнорування цієї відмінності стає підґрунтям для прийняття продуктових рішень, що спираються на статистичний артефакт, а не на реальний ефект.

Тісно пов'язаною проблемою є p-hacking — практика багаторазової перевірки статистичної значущості результатів у процесі накопичення даних без коригування рівня значущості. У цифрових продуктах це явище набуває форми так званого «підглядання» (peeking): аналітик або продакт-менеджер перевіряє результати тесту щодня і зупиняє його, щойно p-value опускається нижче порогу 0,05, не очікуючи досягнення заздалегідь визначеного розміру вибірки. Математично доведено, що при такому підході ймовірність хибнопозитивного результату суттєво перевищує номінальний рівень значущості α: за даними симуляційних досліджень, багаторазова перевірка результатів у процесі накопичення даних може підвищити реальну ймовірність помилки першого роду до 20–30% при номінальному $\alpha = 0{,}05$. Відсутність автоматизованого контролю за дотриманням умови фіксованого горизонту є суттєвим недоліком більшості доступних аналітичних платформ.

Окремою і часто недооціненою проблемою є Sample Ratio Mismatch (SRM) — невідповідність між фактичним та очікуваним співвідношенням користувачів між групами тесту. В ідеальному A/B тесті з рівномірним розподілом трафіку частки груп A і B мають відповідати заздалегідь визначеному цільовому розподілу, наприклад 50/50. Якщо фактичне співвідношення суттєво відхиляється від очікуваного, це свідчить про наявність системної проблеми у механізмі рандомізації або в інфраструктурі збору даних. Причинами SRM можуть бути вибіркова фільтрація ботів лише з однієї групи, кешування відповідей на рівні CDN, помилки в логіці призначення варіантів або втрата подій внаслідок мережевих проблем. Принципово важливо, що навіть незначне SRM здатне повністю анулювати рандомізацію та зробити будь-які

статистичні висновки недостовірними: дві групи більше не є статистично еквівалентними, і порівняння між ними втрачає сенс [5].

Проблема множинних порівнянь виникає, коли в межах одного A/B тесту одночасно аналізується велика кількість метрик або сегментів без відповідного коригування рівня значущості. За умови тестування k незалежних гіпотез при рівні значущості α ймовірність отримати хоча б один хибнопозитивний результат серед усіх k тестів дорівнює $1 - (1 - \alpha)^k$. При k = 20 метриках та α = 0,05 ця ймовірність наближається до 64%, тобто у двох з трьох тестів буде виявлено щонайменше один «значущий» результат, який насправді є випадковим. У практиці цифрових продуктів ця проблема особливо актуальна при сегментному аналізі: перевірка результатів у 10–15 підгрупах без коригування Бонферроні або процедури Бенджаміні-Хохберга майже гарантовано дасть хибнозначущі результати по окремих сегментах [6].

Ефект новизни (novelty effect) являє собою тимчасове викривлення поведінки користувачів, спричинене їхньою реакцією на саму новизну зміни, а не на її змістовну цінність. Користувачі, яким показано оновлений інтерфейс мобільного застосунку або нову функціональність SaaS-продукту, можуть демонструвати підвищену активність та конверсію у перші дні тесту виключно через зацікавленість новим, тоді як після звикання їхня поведінка повертається до базового рівня. Якщо тест завершується у цей початковий період, виміряний ефект буде суттєво завищеним і не відтвориться після повноцінного впровадження зміни. Аналогічна проблема — у протилежному напрямку — виникає при тривалих тестах, що охоплюють кілька тижнів: сезонні коливання активності аудиторії (вихідні дні, свята, маркетингові акції) можуть систематично впливати на одну групу більше, ніж на іншу, якщо розподіл трафіку у часі нерівномірний.

Гетерогенність ефекту (Heterogeneous Treatment Effect, HTE) є проблемою, що виникає, коли середній ефект тесту по всій аудиторії маскує суттєво різні

ефекти у різних підгрупах користувачів. Наприклад, зміна алгоритму рекомендацій у мобільному застосунку може давати значний позитивний uplift серед досвідчених користувачів із тривалою історією взаємодії, але при цьому негативно впливати на нових користувачів, які ще не сформували стійких уподобань. Якщо система аналізу не підтримує сегментного аналізу з коректним контролем за множинними порівняннями, ця неоднорідність залишається прихованою, і рішення про впровадження зміни ухвалюється на основі усередненого ефекту, що не відображає реального впливу на жодну з ключових підгруп аудиторії.

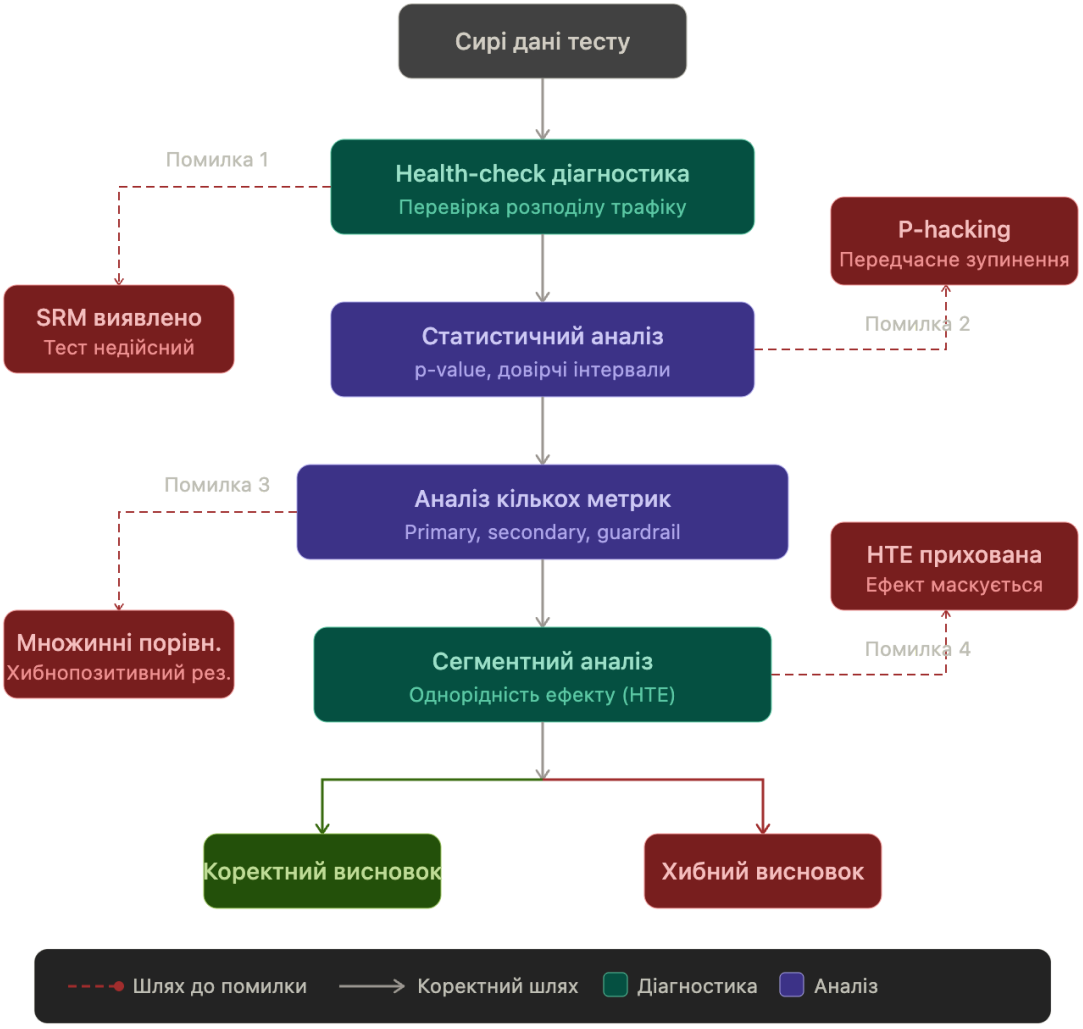


Рисунок 1.1 – Типові джерела помилок при інтерпретації результатів A/B тесту(діаграма: від сирих даних через SRM → p-hacking → множинні порівняння → HTE → хибний висновок)

Схему типових помилок при аналізі A/B тесту наведено на рисунку 1.1. Діаграма ілюструє ключові точки, в яких у процесі аналізу можуть виникати системні помилки: на етапі перевірки якості даних (SRM), на етапі статистичного тестування (p-hacking, множинні порівняння) та на етапі інтерпретації ефекту (ефект новизни, HTE). Жодна з комерційних платформ,

розглянутих у попередньому підрозділі, не забезпечує автоматизованої діагностики по всіх зазначених точках одночасно.

Сукупність описаних проблем свідчить про те, що коректна інтерпретація результатів A/B тесту є складнішою задачею, ніж простий розрахунок p-value та порівняння середніх. Вона потребує комплексної діагностики якості даних, контролю за умовами проведення тесту, перевірки однорідності ефекту по сегментах та зваженої агрегації сигналів від кількох метрик. Саме ці вимоги формують функціональний базис для розробки спеціалізованої системи підтримки прийняття рішень, постановка задачі та конкретні вимоги до якої викладені у наступному підрозділі.

1.5 Постановка задачі та вимоги до системи

Аналіз предметної області, проведений у попередніх підрозділах, дозволяє сформулювати чітку постановку задачі даної роботи. Встановлено, що A/B тестування є ключовим методом підтримки продуктових рішень у цифрових продуктах, однак його практичне застосування супроводжується комплексом систематичних проблем: від структурних вад даних (SRM) до некоректної статистичної інтерпретації та прихованої неоднорідності ефекту по сегментах аудиторії. Існуючі платформи не забезпечують комплексного вирішення цих проблем в єдиному аналітичному середовищі, доступному для широкого кола продуктових команд без значних інженерних ресурсів. Це визначає мету та предмет розробки даної роботи.

Метою роботи є проектування та реалізація системи підтримки прийняття рішень (СПР) для аналізу результатів A/B тестування у цифрових продуктах, яка автоматизує повний аналітичний цикл — від завантаження та валідації сирих експериментальних даних до формування зваженої аналітики щодо впровадження тестової зміни. Предметом розробки є веб застосунок, що

реалізує статистичний аналіз, CI-оцінювання, сегментацію користувачів, health-check діагностику та механізм агрегованих рекомендацій на основі системи первинних, вторинних та охоронних метрик.

Цільовою аудиторією системи є продакт-менеджери, продуктові аналітики та команди зростання (growth teams) компаній, що розробляють мобільні застосунки, SaaS-продукти та вебсервіси. Ключовою вимогою з боку цієї аудиторії є доступність аналітичних результатів без необхідності глибоких знань математичної статистики: система має не лише виконувати розрахунки, а й представляти їх у формі, зрозумілій для прийняття бізнес-рішень. Водночас аналітики з технічною підготовкою повинні мати доступ до деталізованих статистичних показників для самостійної верифікації результатів.

Функціональні вимоги до системи формуються безпосередньо з проблем, виявлених у підрозділі 1.4, і забезпечують їх системне вирішення. Першою функціональною вимогою є підтримка завантаження та автоматизованої валідації CSV-файлів з експериментальними даними: система має перевіряти наявність обов'язкових полів (ідентифікатор користувача, поле варіанта), визначати типи метрик та виявляти структурні аномалії у даних ще до початку статистичного аналізу. Друга вимога — розрахунок повного набору статистичних показників: conversion rate, абсолютного та відносного uplift, p-value та двосторонніх довірчих інтервалів для кожної з обраних метрик.

Третя функціональна вимога стосується реалізації bootstrap-аналізу як альтернативного методу оцінки довірчих інтервалів для метрик з асиметричним розподілом. Це особливо актуально для метрик доходу (ARPU, LTV) у SaaS-продуктах та мобільних застосунках, де parametric confidence intervals, що базуються на нормальному розподілі, дають ненадійні оцінки. Четвертою вимогою є автоматизована health-check діагностика: система має виявляти Sample Ratio Mismatch шляхом порівняння фактичного розподілу трафіку між групами з очікуваним та генерувати попередження при виявленні суттєвих

відхилень. П'ята вимога — реалізація сегментованого аналізу на основі user properties, що дозволяє виявляти гетерогенність ефекту серед різних підгруп аудиторії.

Шоста функціональна вимога є системоутворюючою з точки зору підтримки прийняття рішень: система має підтримувати структуровану роботу з трьома типами метрик — primary, secondary та guardrail — та формувати структурований аналітичний репорт, що відображає стан усіх трьох категорій і включає попередження при виявленні значущого погіршення guardrail-метрик. Зокрема, за наявності позитивного статистично значущого ефекту на первинну метрику одночасно із значущим погіршенням guardrail-показника система має фіксувати це як конфліктний сигнал та відображати відповідне попередження в репорті. Остаточне рішення щодо впровадження залишається за аналітиком на основі наданих даних. Відповідність між виявленими проблемами, функціональними вимогами та модулями системи наведено у таблиці 1.3.

Таблиця 1.3 – Відповідність виявлених проблем функціональним вимогам до системи

Виявлена проблема	**Функціональна вимога**	**Модуль системи**
Некоректна інтерпретація p-value	Розрахунок p-value з поясненням та confidence intervals	Статистичний аналіз
Sample Ratio Mismatch	Автоматизована SRM-перевірка з попередженням	Health-check аналіз
Множинні порівняння метрик	Розмежування primary, secondary, guardrail метрик	Управління метриками
Гетерогенність ефекту (HTE)	Сегментований аналіз за user properties	Сегментація користувачів
Асиметрія розподілу метрик	Bootstrap-оцінка confidence intervals	Bootstrap аналіз

Нефункціональні вимоги до системи визначають якісні характеристики її реалізації. Система має бути реалізована як веб застосунок, доступний без встановлення додаткового програмного забезпечення, що забезпечує її доступність для широкого кола спеціалістів незалежно від використовуваної операційної системи. Вхідним форматом даних обрано CSV — найбільш поширений формат експорту даних з аналітичних платформ та сховищ даних, що використовуються у цифрових продуктах. Інтерфейс системи має забезпечувати інтерактивність аналізу: можливість змінювати склад метрик та параметри сегментації без повторного завантаження файлу з даними.

Таким чином, сформульована задача передбачає розробку інтегрованого аналітичного інструменту, що поєднує в собі функції статистичного аналізу, діагностики якості експерименту та підтримки прийняття рішень в єдиному веб інтерфейсі. Математичний апарат, методи та моделі, що лежать в основі кожного

функціонального модуля системи, є предметом детального розгляду у другому розділі роботи.

1.6 Висновки до розділу 1

У першому розділі роботи проведено комплексний аналіз предметної області, що охоплює теоретичні засади A/B тестування, систему метрик продуктової аналітики, стан існуючих платформ та типові проблеми інтерпретації результатів онлайн-експериментів у цифрових продуктах. На основі цього аналізу сформульовано постановку задачі та визначено вимоги до розроблюваної системи.

Встановлено, що A/B тестування є методом рандомізованого контрольованого експерименту, адаптованого для цифрового середовища, і слугує основним інструментом емпіричного обґрунтування продуктових рішень у мобільних застосунках, SaaS-продуктах та вебсервісах. Коректне проведення тесту передбачає дотримання принципів рандомізації, ізоляції між групами та репрезентативності вибірки, а математичну основу становить апарат статистичного тестування гіпотез з визначенням нульової та альтернативної гіпотез, рівня значущості та потужності тесту.

Визначено, що ефективний аналіз A/B тесту потребує структурованої системи метрик, яка включає первинні метрики як критерій рішення, вторинні метрики для поглибленої інтерпретації ефекту та охоронні (guardrail) метрики для захисту від небажаних побічних наслідків тестованої зміни. Показано, що conversion rate та uplift є центральними показниками аналізу, а наявність асиметричних безперервних метрик (доход, тривалість сесії) зумовлює необхідність застосування непараметричних методів оцінювання.

Огляд сучасних платформ для A/B тестування виявив суттєвий функціональний розрив між комерційними рішеннями та реальними потребами продуктових команд. Комерційні платформи (Optimizely, VWO) обмежені у статистичних методах і не підтримують bootstrap-аналізу та автоматизованих health-check перевірок. Відкриті рішення (GrowthBook, PlanOut) потребують

значної інженерної інфраструктури. Внутрішні системи технологічних лідерів (Airbnb, Netflix), попри функціональну повноту, є недоступними для широкого загалу.

Систематизовано типові проблеми інтерпретації результатів A/B тестів: некоректна інтерпретація p-value, p-hacking та передчасне завершення тесту, Sample Ratio Mismatch як індикатор структурних проблем у даних, ефект множинних порівнянь при аналізі кількох метрик і сегментів, ефект новизни та сезонні коливання, а також гетерогенність ефекту по підгрупах аудиторії. Показано, що жодна з доступних платформ не забезпечує автоматизованої діагностики по всіх зазначених проблемах одночасно.

На підставі виявлених проблем сформульовано постановку задачі та визначено функціональні вимоги до системи підтримки прийняття рішень: автоматизована валідація CSV-даних, розрахунок повного набору статистичних показників, bootstrap-оцінювання довірчих інтервалів, SRM-діагностика у межах health-check модуля, сегментований аналіз за user properties та формування структурованого аналітичного репорту з попередженнями про конфліктні сигнали між метриками. Математичні моделі та методи, що забезпечують реалізацію кожного з цих модулів, є предметом розгляду у другому розділі роботи.

РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ АНАЛІЗУ A/B ТЕСТІВ

2.1 Статистичні підходи до аналізу експериментів

Статистичний аналіз є ядром будь-якої системи оцінки результатів A/B тестування, оскільки саме він забезпечує перехід від спостережуваних відмінностей між групами до обґрунтованих висновків про наявність або відсутність реального ефекту. Вибір статистичного підходу визначає не лише точність оцінок, а й характер рекомендацій, які система здатна формувати. У контексті аналізу онлайн-експериментів у цифрових продуктах найбільшого поширення набули два принципово різних підходи: частотний (frequentist) та байєсівський (Bayesian). Кожен з них спирається на різні концептуальні засади та по-різному відповідає на питання про інтерпретацію результатів тесту.

Частотний підхід, що є домінуючим у промисловій практиці A/B тестування, ґрунтується на логіці перевірки нульової гіпотези. Відповідно до цього підходу, перед початком тесту формулюється нульова гіпотеза H_0, яка стверджує відсутність різниці між контрольною та тестовою групами за досліджуваною метрикою, та альтернативна гіпотеза H_1, яка стверджує наявність такої різниці. Після збору даних обчислюється тестова статистика, що вимірює відхилення спостережуваних результатів від тих, що очікувалися б за умови істинності H_0. На підставі цієї статистики та відомого теоретичного розподілу визначається p-value — ймовірність отримати спостережуване або більш екстремальне значення тестової статистики за умови, що нульова гіпотеза є істинною. Якщо p-value не перевищує заздалегідь встановленого порогу значущості α, нульова гіпотеза відхиляється на користь альтернативної [7].

Для порівняння часток (conversion rate) між двома групами стандартним інструментом є z-тест для двох пропорцій. Тестова статистика z обчислюється за формулою:

$$z = \frac{\hat{p}_B - \hat{p}_A}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_A}+\frac{1}{n_B})}}, \quad (2.1)$$

де $\hat{p}_A$ та $\hat{p}_B$ — вибіркові частки (conversion rate) у контрольній та тестовій групах відповідно;

n_A та n_B — обсяги відповідних груп;

p̂ — об'єднана вибіркова частка, що розраховується як $\hat{p} = \frac{(x_A+x_B)}{(n_A+n_B)}$, x_A та x_B — кількість конверсій у кожній групі.

За умови достатнього обсягу вибірки статистика z має стандартний нормальний розподіл, що дозволяє визначити p-value через таблиці або функцію розподілу Φ(z). Для безперервних метрик, таких як дохід або тривалість сесії, замість z-тесту застосовується t-тест Велча, що не потребує припущення про рівність дисперсій у двох групах.

Важливим поняттям частотного підходу є довірчий інтервал (confidence interval, CI) — діапазон значень, що з заданою ймовірністю (1 − α) містить справжнє значення параметра популяції. Для різниці двох пропорцій $\Delta = p_B - p_A$ двосторонній довірчий інтервал рівня (1 − α) має вигляд:

$$CI = (\hat{p}_B - \hat{p}_A) \pm z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{\hat{p}_A(1-\hat{p}_A)}{n_A} + \frac{\hat{p}_B(1-\hat{p}_B)}{n_B}}, \quad (2.2)$$

де $z_{\frac{\alpha}{2}}$ — квантиль стандартного нормального розподілу рівня $\frac{\alpha}{2}$ (для $\alpha = 0{,}05$ значення $z_{\frac{\alpha}{2}} \approx 1{,}96$).

Довірчий інтервал є більш інформативним показником, ніж p-value, оскільки відображає не лише факт статистичної значущості, а й діапазон правдоподібних значень ефекту. Якщо довірчий інтервал для різниці пропорцій не містить нуля, це еквівалентно відхиленню нульової гіпотези при відповідному рівні значущості. У системі, що розробляється, обидва показники — p-value та CI — відображаються спільно, що дозволяє аналітику оцінювати як статистичну значущість, так і практичну величину виявленого ефекту.

Щодо вибору між одностороннім та двостороннім тестом: двосторонній тест перевіряє гіпотезу про будь-яку відмінність між групами (як позитивну, так і негативну), тоді як односторонній тест спрямований на виявлення ефекту лише в одному напрямку. У практиці цифрових продуктів рекомендованим є двосторонній тест: він є більш консервативним та захищає від помилкового висновку у випадках, коли тестована зміна несподівано погіршує метрику. Використання одностороннього тесту виправдане лише за наявності сильного апріорного обґрунтування напряму ефекту, що у більшості реальних продуктових гіпотез не виконується.

Байєсівський підхід пропонує концептуально відмінну інтерпретацію результатів тесту. Замість p-value він оперує апостеріорним розподілом параметра ефекту, що формується шляхом поєднання апріорних уявлень аналітика про можливий ефект із правдоподібністю спостережуваних даних. Ключовим результатом байєсівського аналізу є не бінарне рішення «відхилити / не відхилити H_0», а розподіл ймовірностей можливих значень ефекту, а також показник «Probability to Beat Baseline» — ймовірність того, що варіант B перевершує варіант A. Ця інтерпретація є більш природною для продуктових рішень, оскільки надає аналітику кількісну оцінку впевненості у перевазі

тестованого варіанта, а не лише бінарний висновок. Однак байєсівський підхід суттєво чутливіший до вибору апріорного розподілу, що ускладнює його стандартизоване застосування у системах автоматизованого аналізу [8].

У контексті проблеми множинних порівнянь, описаної у підрозділі 1.4, частотний підхід потребує застосування процедур коригування рівня значущості. Найпростішою є поправка Бонферроні: при одночасному тестуванні k гіпотез скоригований рівень значущості для кожної з них встановлюється на рівні $\alpha' = \frac{\alpha}{k}$. Ця поправка є консервативною і може призводити до підвищення кількості хибнонегативних результатів. Менш консервативною альтернативою є процедура Бенджаміні-Хохберга, що контролює не ймовірність хоча б однієї помилки (Family-Wise Error Rate), а частку хибнопозитивних результатів серед всіх значущих (False Discovery Rate). Для типової задачі аналізу кількох метрик у A/B тесті процедура Бенджаміні-Хохберга забезпечує кращий баланс між чутливістю та специфічністю [9].

Статистичний апарат аналізу A/B тестів включає сукупність взаємопов'язаних методів: від вибору тестової статистики відповідно до типу метрики до коригування рівня значущості при множинних порівняннях та оцінки практичної значущості виявленого ефекту. Реалізація цього апарату в системі підтримки прийняття рішень забезпечує коректну статистичну основу для подальших модулів — оцінки метрик та ефектів, сегментації та формування рекомендацій, методи яких розглядаються в наступних підрозділах.

2.2 Методи оцінки метрик та ефектів (uplift, conversion)

Статистичне тестування гіпотез, розглянуте у попередньому підрозділі, дає відповідь на запитання про те, чи є різниця між групами статистично значущою. Однак для підтримки продуктового рішення не менш важливо кількісно оцінити сам ефект: наскільки саме змінилась метрика, наскільки точною є ця оцінка та наскільки стабільною вона є відносно природної варіативності даних. Саме ці питання вирішують методи точкової та інтервальної оцінки метрик, а також непараметричні методи оцінювання — зокрема, bootstrap, що є ключовим інструментом для роботи з асиметричними розподілами у цифрових продуктах.

Точкова оцінка conversion rate для групи g визначається як вибіркова частка конверсій і була формалізована у підрозділі 1.2. У контексті A/B тесту центральним об'єктом оцінювання є різниця між conversion rate контрольної та тестової груп — абсолютний ефект $\Delta = \hat{p}_B - \hat{p}_A$ — та відносний uplift, нормований відносно базового рівня контрольної групи. Інтервальна оцінка цих показників за допомогою параметричних довірчих інтервалів, побудованих на основі нормального наближення, є коректною лише за виконання умов: достатньо великого обсягу вибірки (зазвичай n_A, $n_B \geq 100$) та симетричного розподілу метрики. Для бінарних метрик (conversion rate) ці умови, як правило, виконуються при достатньому трафіку. Для безперервних метрик вони часто порушуються.

Безперервні метрики цифрових продуктів — дохід на користувача (ARPU), середня вартість замовлення (AOV), тривалість сесії — характеризуються суттєво правосторонньою асиметрією розподілу. Невелика частка користувачів генерує непропорційно великі значення метрики, що спричиняє значні відхилення від нормального розподілу та робить parametric confidence intervals ненадійними. Застосування стандартного t-тесту у таких умовах може

призводити до занижених або завищених значень стандартної похибки та, відповідно, до некоректної оцінки ширини довірчого інтервалу. Саме ця обставина зумовлює необхідність застосування bootstrap як непараметричного методу, що не спирається на жодні припущення щодо форми розподілу даних.

Bootstrap є методом статистичного оцінювання, що ґрунтується на принципі повторної вибірки із заміщенням (resampling with replacement) [10]. Ідея методу полягає в тому, що наявна вибірка є найкращим наближенням до генеральної сукупності, тому розподіл статистики інтересу можна апроксимувати шляхом багаторазового розрахунку цієї статистики на псевдо вибірках, сформованих з вихідних даних шляхом вибору з поверненням. Процедура bootstrap для оцінки довірчого інтервалу uplift між групами A і B включає такі кроки: з вибірки групи A розміром n_A формується B псевдовибірок того самого розміру шляхом вибору з поверненням; аналогічно для групи B; для кожної з B пар псевдовибірок розраховується значення uplift; з отриманого емпіричного розподілу B значень uplift вилучаються квантилі рівня α/2 та 1 − α/2, що формують межі довірчого інтервалу. Перцентильний bootstrap-довірчий інтервал рівня (1 − α) для параметра θ записується як:

$$CI_{boot} = \left[\hat{\theta}^{*}_{\frac{\alpha}{2}}, \hat{\theta}^{*}_{\frac{1-\alpha}{2}}\right], \tag{2.3}$$

де $\hat{\theta}^{*}_{\frac{\alpha}{2}}$ та $\hat{\theta}^{*}_{\frac{1-\alpha}{2}}$ — відповідні вибіркові квантилі емпіричного розподілу bootstrap-оцінок статистики θ;

B — кількість bootstrap-ітерацій, що зазвичай становить від 1000 до 10000 для забезпечення стабільності оцінки.

На практиці для аналізу revenue-метрик у SaaS-продуктах та мобільних застосунках рекомендованим є значення B = 1000–5000, що забезпечує прийнятну точність при розумному обчислювальному навантаженні [11].

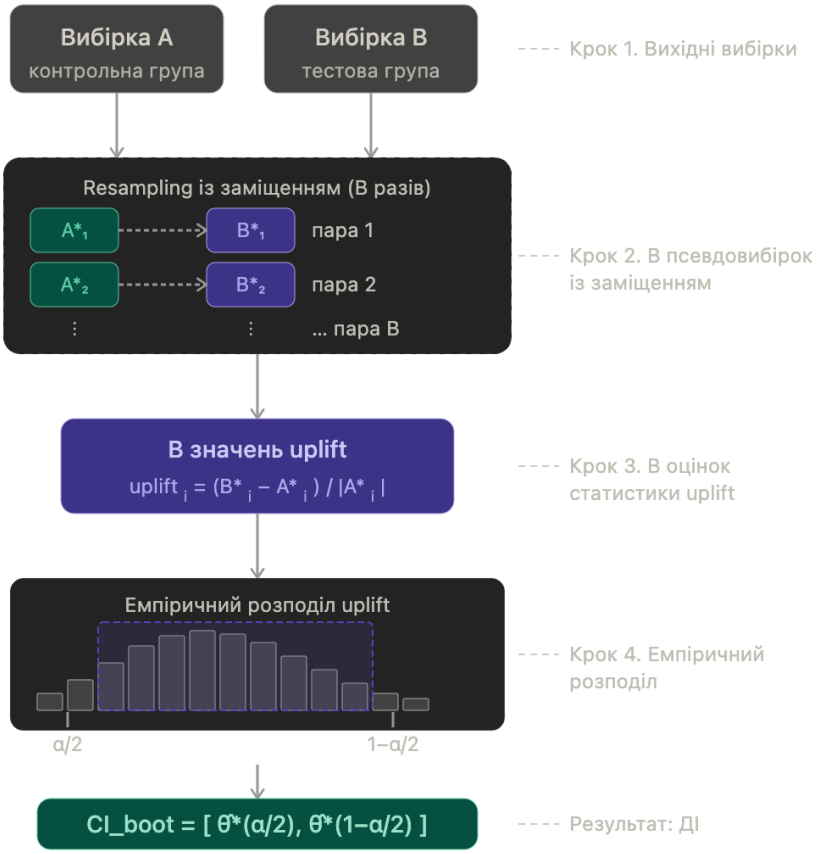


Рисунок 2.1 – Схема процедури bootstrap для оцінки довірчого інтервалу uplift(вихідна вибірка → B повторних вибірок → B оцінок uplift → емпіричний розподіл → ДІ)

Схему процедури bootstrap наведено на рисунку 2.1. Діаграма ілюструє покроковий процес: від вихідних вибірок груп A і B через формування B псевдовибірок та розрахунок B значень статистики до побудови емпіричного розподілу і виокремлення перцентильного довірчого інтервалу. Ключовою перевагою bootstrap є те, що форма отриманого емпіричного розподілу

автоматично відображає реальну асиметрію та важкі хвости вихідних даних, не нав'язуючи нормального наближення [12].

Для складених метрик, що обчислюються як відношення двох величин (наприклад, дохід на сесію = загальний дохід / кількість сесій), параметричні методи оцінювання потребують застосування дельта-методу. Дельта-метод дозволяє наближено оцінити дисперсію функції від вибіркових оцінок через часткові похідні цієї функції. Нехай метрика θ визначається як відношення двох агрегатів: $\hat{\theta} = \frac{\mu_Y}{\mu_X}$, де μ_Y та μ_X — середні значення чисельника і знаменника відповідно. Тоді дисперсія оцінки $\hat{\theta}$ апроксимується за дельта-методом як:

$$Var(\hat{\theta}) \approx \frac{1}{\hat{\mu}_X^2}\left[Var(\overline{Y}) + \hat{\theta}^2 Var(\overline{X}) - 2\hat{\theta}Cov(\overline{Y},\overline{X})\right], \quad (2.4)$$

де $Var(\overline{Y})$ та $Var(\overline{X})$ — дисперсії вибіркових середніх чисельника і знаменника;

$Cov(\overline{Y},\overline{X})$ — їхня вибіркова коваріація.

Дельта-метод забезпечує асимптотично коректну оцінку дисперсії складеної метрики і є стандартним інструментом у платформах аналізу A/B тестів, що працюють з revenue per session або подібними показниками. Bootstrap у цьому випадку слугує практичною альтернативою, оскільки автоматично враховує структуру складеної метрики без явного обчислення часткових похідних.

Важливим аспектом оцінки ефектів є відмінність між абсолютним та відносним uplift з точки зору їхньої інтерпретації та застосування у продуктових рішеннях. Абсолютний uplift $\Delta = \hat{p}_B - \hat{p}_A$ безпосередньо відображає зміну метрики у числових одиницях і є більш зрозумілим для нетехнічної аудиторії:

«conversion rate зріс на 0,5 відсоткового пункту». Відносний uplift, виражений у відсотках від базового рівня, є зручнішим для порівняння ефектів між різними тестами та метриками з різними базовими значеннями. Обидва показники несуть важливу інформацію і мають відображатися системою аналізу спільно: абсолютний uplift — для оцінки операційного впливу на бізнес, відносний — для порівняльного аналізу ефективності різних продуктових змін.

Повноцінна оцінка ефектів у A/B тесті включає точкові оцінки conversion rate і uplift для кожної групи, параметричні довірчі інтервали для бінарних метрик при дотриманні умов нормального наближення, bootstrap-інтервали для безперервних та асиметричних метрик, а також дельта-метод для складених показників. Комбінація цих інструментів у єдиній системі забезпечує коректну кількісну оцінку ефекту незалежно від типу та розподілу аналізованої метрики. Питання про те, як ці оцінки можуть відрізнятися між підгрупами аудиторії та яким чином виявити таку неоднорідність, розглядається у наступному підрозділі.

2.3 Методи сегментації користувачів

Аналіз середнього ефекту A/B тесту на рівні всієї аудиторії, описаний у попередніх підрозділах, є необхідним, але недостатнім для повноцінної підтримки продуктового рішення. Реальна аудиторія цифрових продуктів є неоднорідною: різні підгрупи користувачів можуть по-різному реагувати на одну й ту саму зміну у продукті. Це явище, відоме як гетерогенність ефекту лікування (Heterogeneous Treatment Effect, HTE), є поширеним у практиці онлайн-експериментів і означає, що усереднений ефект може маскувати суттєво різні результати у різних сегментах аудиторії. Методи сегментації користувачів дозволяють виявити цю неоднорідність та отримати більш детальну картину впливу тестованої зміни.

З методологічної точки зору сегментація у контексті A/B тестів поділяється на два принципово різних підходи: апріорну та апостеріорну. Апріорна сегментація передбачає визначення підгруп для аналізу до початку тесту або, щонайменше, до перегляду результатів. Сегменти формуються на основі заздалегідь відомих характеристик користувачів — платформи, географії, типу підписки, стажу у продукті тощо. Оскільки ці сегменти визначені незалежно від спостережуваних результатів, статистичний аналіз всередині кожного сегмента залишається коректним за умови відповідного коригування рівня значущості на множинні порівняння. Апостеріорна сегментація, навпаки, передбачає пошук сегментів після перегляду результатів — вибір тих підгруп, для яких ефект виявився найбільш вираженим. Такий підхід є формою data dredging і призводить до суттєвого завищення ймовірності хибнопозитивних результатів, оскільки з достатньої кількості сегментів завжди знайдеться той, що демонструє «значущий» ефект за суто випадкових причин [13].

Практичне значення сегментного аналізу у цифрових продуктах важко переоцінити. Розглянемо характерний приклад: A/B тест у мобільному застосунку перевіряє новий онбординговий потік. Загальний uplift conversion rate виявляється статистично незначущим. Однак сегментний аналіз за ознакою операційної системи показує, що серед користувачів iOS ефект є позитивним і значущим (+8% uplift), тоді як серед користувачів Android — від'ємним (−4% uplift). Ці два ефекти компенсують один одного на рівні загальної аудиторії, і без сегментації цінна інформація про диференційований вплив зміни залишилась би прихованою. Подібна ситуація характерна і для SaaS-продуктів, де ефект на платних та безкоштовних користувачів може суттєво відрізнятися.

Типологію сегментів, що застосовуються в аналізі A/B тестів цифрових продуктів, наведено у таблиці 2.1. Кожен тип сегменту відповідає певній аналітичній меті та дозволяє відповісти на специфічне продуктове запитання.

Таблиця 2.1 – Типи сегментів у A/B аналізі цифрових продуктів

Тип сегменту	**Ознака поділу**	**Приклад у цифровому продукті**	**Аналітична цінність**
Платформний	Операційна система або пристрій	iOS vs Android у мобільному застосунку	Виявлення платформо-специфічних ефектів
Географічний	Країна або регіон	Ринки з різною платіжною поведінкою в SaaS	Адаптація продукту до локальних ринків
Поведінковий	Стаж та активність користувача	Нові vs повернені користувачі вебсервісу	Виявлення ефекту новизни, HTE
Тарифний	Тип підписки або план	Free vs Pro у SaaS-продукті	Оцінка ефекту для монетизованої аудиторії
Канальний	Джерело трафіку	Органічний vs платний трафік	Диференціація за якістю аудиторії

З операційної точки зору реалізація сегментованого аналізу в системі передбачає такий алгоритм: після завантаження CSV-файлу з даними тесту система автоматично ідентифікує категоріальні колонки як потенційні ознаки сегментації. Для кожного обраного сегментаційного атрибута дані розбиваються на підгрупи за унікальними значеннями відповідної колонки, після чого для кожної підгрупи незалежно виконується повний статистичний аналіз: розрахунок conversion rate, uplift, p-value та довірчих інтервалів. Результати по всіх сегментах зводяться у порівняльну таблицю, що дозволяє аналітику виявити підгрупи з аномально високим або низьким ефектом.

Статистична коректність сегментного аналізу вимагає обов'язкового коригування рівня значущості на множинні порівняння. Якщо аналізується s сегментів за одним атрибутом, кожен з яких тестується при рівні значущості α, сімейний рівень помилки (Family-Wise Error Rate, FWER) без коригування може суттєво перевищувати α. Для контролю FWER застосовується поправка Бонферроні зі скоригованим рівнем $\alpha' = \frac{\alpha}{s}$. Альтернативою є контроль частки хибних відкриттів (False Discovery Rate, FDR) за процедурою Бенджаміні-Хохберга, що є менш консервативним підходом і більше підходить для розвідувального аналізу з великою кількістю сегментів. Вибір між цими підходами залежить від того, чи є сегментний аналіз підтверджувальним (confirmatory), де коректність кожного висновку критична, чи розвідувальним (exploratory), де більш важливою є висока чутливість.

Окрему методологічну увагу заслуговує питання мінімального розміру сегмента, необхідного для статистично надійного аналізу. Для малих сегментів обсяг вибірки може бути недостатнім для виявлення ефектів навіть тих розмірів, що є практично значущими. У такому разі довірчі інтервали виявляються надмірно широкими, а p-value не досягає порогового рівня навіть при реально існуючому ефекті — виникає проблема недостатньої статистичної потужності. Практичним орієнтиром є вимога, щоб кожен підсегмент містив щонайменше ту кількість спостережень, що відповідає мінімальному розміру вибірки для заданих α та $1-\beta$, розрахованому для очікуваного розміру ефекту. Якщо ця умова не виконується, система має відображати відповідне попередження, запобігаючи інтерпретації статистично ненадійних результатів.

Health-check аналіз балансу сегментів є невід'ємною складовою сегментованого аналізу. Окрім перевірки загального Sample Ratio Mismatch між групами A і B, система має перевіряти розподіл ключових сегментаційних атрибутів у межах кожної групи. Якщо, наприклад, частка iOS-користувачів у групі A становить 60%, а в групі B — 45%, це свідчить про порушення

рандомізації, що могло виникнути внаслідок системних помилок у механізмі розподілу трафіку. Таке порушення балансу сегментів анулює коректність будь-якого сегментного аналізу і є важливим сигналом для аналітика ще до переходу до інтерпретації результатів. Поєднання сегментації з health-check діагностикою формує комплексний підхід до аналізу якості та результатів A/B тесту, що є основою для підтримки прийняття рішень, концепцію якої розглянуто у наступному підрозділі.

2.4 Концепція систем підтримки прийняття рішень

Методи статистичного аналізу, оцінки ефектів та сегментації, розглянуті у попередніх підрозділах, утворюють аналітичне ядро системи. Однак самі по собі числові результати — p-value, uplift, confidence interval, розподіл по сегментах — ще не є відповіддю на продуктове запитання. Перетворення цього набору обчислень на структурований аналітичний звіт, зручний для інтерпретації та прийняття рішення, потребує додаткового рівня організації. Цей рівень реалізується через концепцію системи підтримки прийняття рішень (СПР, англ. Decision Support System, DSS) — класу інформаційних систем, орієнтованих на підвищення якості та обґрунтованості управлінських рішень в умовах складності та часткової невизначеності.

Концепція DSS сформувалася у 1970-х роках у роботах Горрі та Мортона, Кіна та Скотт-Мортона як відповідь на обмеження традиційних управлінських інформаційних систем, що надавали лише агреговані звіти про стан справ, але не підтримували процес аналізу та інтерпретації. Класична архітектура DSS включає три взаємопов'язані підсистеми. Перша — підсистема управління даними, що забезпечує доступ до релевантних даних, їх зберігання та попередню обробку. Друга — підсистема управління моделями, що містить набір аналітичних методів, застосовуваних для обробки даних та генерації аналітичних висновків. Третя — підсистема інтерфейсу користувача, що

представляє результати у структурованому та зручному для інтерпретації вигляді.

Принципово важливою відмінністю DSS від систем автоматичного прийняття рішень є збереження за людиною ролі кінцевого суб'єкта рішення. DSS не замінює аналітика або продакт-менеджера і не формує за нього готовий висновок — вона структурує аналітичні дані таким чином, щоб особа, яка приймає рішення, могла швидко охопити всю релевантну інформацію, зіставити сигнали від різних метрик та сегментів, виявити потенційні проблеми з якістю даних і на цій основі самостійно дійти обґрунтованого висновку. Ця властивість є особливо значущою в контексті продуктової аналітики, де рішення нерідко вимагають інтеграції кількісних результатів тесту з якісним розумінням продуктового контексту, стратегічних пріоритетів та обмежень, що перебувають поза межами будь-якої формалізованої моделі.

Застосування концепції DSS до задачі аналізу A/B тестів у цифрових продуктах породжує специфічну архітектуру, що відображає особливості предметної області. Підсистема управління даними реалізується як модуль завантаження та валідації CSV-файлів з результатами експерименту: вона забезпечує перевірку структури вхідних даних, ідентифікацію типів метрик і сегментаційних атрибутів та підготовку даних до аналітичної обробки. Підсистема управління моделями включає статистичні модулі (z-тест для пропорцій, t-тест Велча, bootstrap), модуль health-check діагностики (перевірка Sample Ratio Mismatch, аналіз балансу сегментів) та модуль сегментованого аналізу. Підсистема інтерфейсу реалізується як інтерактивний вебзастосунок, що представляє зведений структурований репорт із результатами по всіх метриках, попередженнями та сегментним розбиттям [14].

Ключовим принципом побудови інтерфейсного шару розробленої системи є агрегація різнорідних аналітичних сигналів у єдиний структурований репорт, де результати по кожній метриці відображаються паралельно із зазначенням її

типу (primary, secondary або guardrail) та статусу (статистично значуще поліпшення, значуще погіршення, відсутність значущих змін). Такий формат подачі дозволяє аналітику або продакт-менеджеру у рамках одного перегляду оцінити загальну картину експерименту: чи підтверджує тест гіпотезу за основною метрикою, чи не порушуються guardrail-показники, які сегменти демонструють нетипову поведінку. Ці дані є необхідною і достатньою основою для прийняття рішення — але саме рішення залишається за людиною.

Суттєвою складовою репорту є блок попереджень, що формується за результатами health-check діагностики та аналізу guardrail-метрик. Попередження про виявлений Sample Ratio Mismatch сигналізує про можливі структурні проблеми у даних, що ставлять під сумнів достовірність будь-яких статистичних висновків. Попередження про значуще погіршення guardrail-метрики за наявності позитивного ефекту на primary метрику вказує на суперечливість результатів і необхідність більш детального аналізу перед прийняттям рішення. Попередження про малий розмір сегмента фіксує недостатню статистичну потужність для надійної інтерпретації сегментного результату. Всі ці сигнали відображаються структуровано — поруч із відповідними числовими результатами, а не окремо від них — що дозволяє аналітику одночасно бачити і сам показник, і контекст його інтерпретації.

Важливою властивістю системи є інтерактивність аналізу: можливість змінювати склад метрик та параметри сегментації без повторного завантаження файлу з даними. Після зміни параметрів система автоматично перераховує всі залежні показники та оновлює репорт у режимі реального часу. Ця властивість підтримує ітеративний аналітичний процес, при якому аналітик послідовно перевіряє різні гіпотези щодо природи спостережуваного ефекту — наприклад, звужує аналіз до конкретного сегмента, змінює первинну метрику або виключає окремі guardrail-показники — не вдаючись до написання додаткового коду або повторного запуску обчислень.

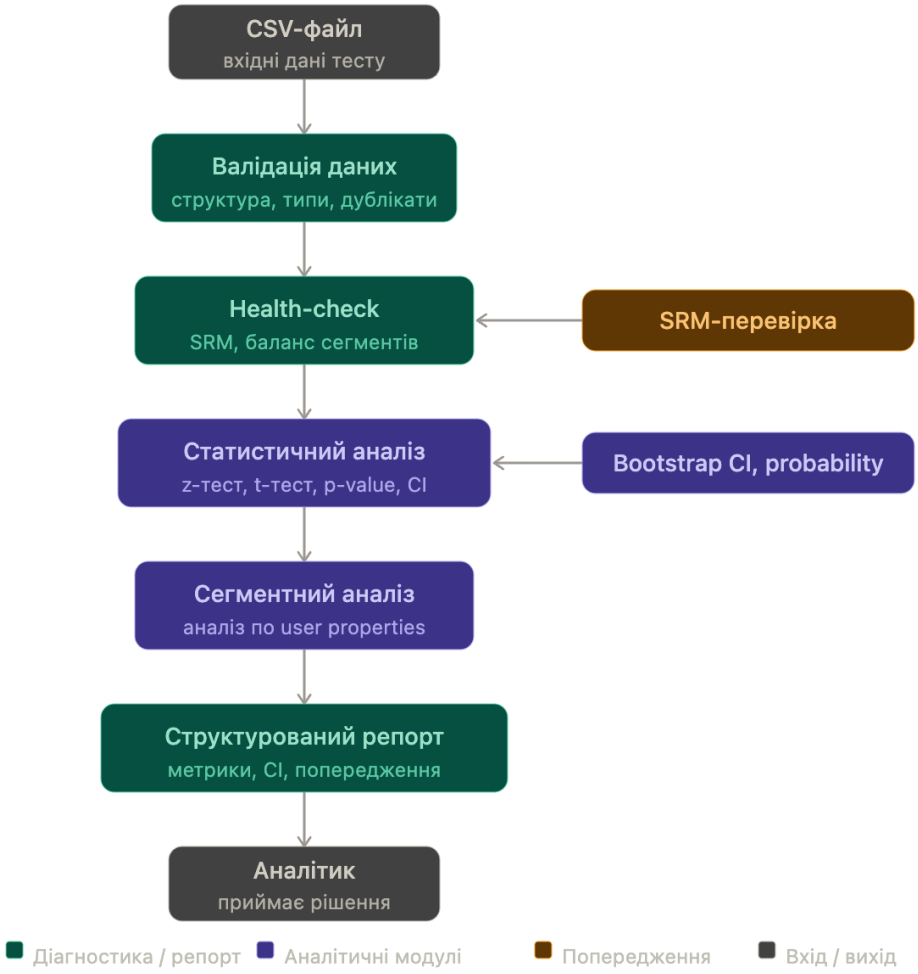


Рисунок 2.2 – Архітектура DSS для аналізу A/B тестів

Загальну архітектуру розробленої системи наведено на рисунку 2.2. Блок-схема відображає послідовність обробки даних від завантаження CSV-файлу через модулі валідації та health-check до статистичного аналізу і сегментації. Результати всіх модулів агрегуються у структурований репорт, що є кінцевим виходом системи та основою для прийняття рішення аналітиком. Стрілка від репорту до аналітика — а не до автоматичного висновку — відображає принципову архітектурну властивість системи: вона розширює аналітичні можливості людини, а не підміняє її судження.

У циклі продуктового розвитку DSS займає позицію між етапом завершення збору даних і етапом прийняття рішення командою. На вхід вона отримує сирі дані завершеного A/B тесту, а на виході формує структурований аналітичний репорт: статистичні оцінки ефектів по всіх метриках з розбивкою на типи, результати health-check діагностики, сегментний аналіз із контролем за множинними порівняннями та блок попереджень про виявлені аномалії. Таким чином, система вбудовується в існуючий продуктовий процес як аналітичний інструмент, що скорочує час від завершення тесту до прийняття рішення та підвищує його обґрунтованість за рахунок комплексного охоплення всіх релевантних аналітичних сигналів. Математична формалізація структури репорту та критеріїв формування попереджень є предметом наступного підрозділу.

2.5 Формалізація задачі прийняття рішення

Попередні підрозділи сформували повний методологічний базис для аналізу результатів A/B тестування: статистичні критерії оцінки значущості, методи інтервального оцінювання ефектів, підходи до сегментного аналізу та архітектурну концепцію DSS. Завершальним кроком теоретичного обґрунтування є формалізація того, яким чином результати всіх аналітичних модулів агрегуються в єдиний структурований репорт — кінцевий вихід системи, що надається аналітику для інтерпретації та прийняття продуктового рішення.

Основною одиницею репорту є набір показників по кожній метриці m з множини $M = \{m_1, m_2, ..., m_□\}$, обраних аналітиком перед початком аналізу. Для кожної метрики система обчислює вектор $\varphi(m)$, що включає: кількість спостережень у контрольній і тестовій групах ($n_{control}$, $n_{treatment}$); середнє значення або conversion rate у кожній групі ($value_{control}$, $value_{treatment}$);

відносний uplift; p-value, отримане відповідним статистичним тестом залежно від типу метрики; та probability — ймовірність того, що тестовий варіант перевершує контрольний, розраховану за байєсівським підходом через Beta-розподіл для бінарних метрик або через bootstrap-порівняння середніх для безперервних. Ці показники відповідають полям датакласу MetricAnalysis, реалізованого у системі.

Поряд з вектором φ(m) система обчислює перцентильний bootstrap-довірчий інтервал для uplift кожної метрики. Нижня та верхня межі інтервалу [lo_u, hi_u] розраховуються за формулою (2.3) на основі емпіричного розподілу B bootstrap-оцінок uplift та відображаються у тултипі над відповідним рядком таблиці результатів разом із повним розподілом. Довірчий інтервал несе додаткову аналітичну цінність відносно p-value: він відображає не лише факт статистичної значущості, а й діапазон правдоподібних значень ефекту та ступінь невизначеності в оцінці, що є важливим для інтерпретації результатів при невеликих вибірках або високій варіативності метрики.

Кожна метрика у репорті відображається разом із своєю роллю — primary, secondary або guardrail, — що задається аналітиком до початку аналізу. Роль впливає на візуальне оформлення: рядки таблиці результатів отримують кольорове виділення відповідно до типу метрики, що дозволяє аналітику одним поглядом зорієнтуватися в ієрархії показників. Принципово важливо, що система не агрегує ролі метрик у єдиний висновок і не генерує готове рішення на їх основі — роль є виключно навігаційним маркером у структурі репорту.

Окремим блоком репорту є результати health-check діагностики, що формуються незалежно від статистичних показників метрик. Центральним елементом health-check є перевірка Sample Ratio Mismatch: функція compute_variant_balance_for_srm() обчислює фактичний розподіл користувачів між контрольною і тестовою групами та порівнює відхилення від очікуваного розподілу 50/50 із пороговим значенням HC_SRM_THRESHOLD_PP = 0,05.

Якщо відхилення перевищує поріг, у репорті відображається попередження, яке сигналізує про можливе порушення рандомізації та ставить під сумнів достовірність усіх статистичних показників. Поряд із SRM-перевіркою система через функцію covariate_balance_property() відображає розподіл ключових user properties між групами у вигляді таблиці та bar-chart, що дозволяє аналітику візуально оцінити баланс сегментів і виявити потенційні перекоси, що не охоплюються SRM-метрикою.

Умову відсутності SRM можна формально записати як:

$$srm_{ok} = (max(\left|p_c - 0.5\right|, \left|p_t - 0.5\right|) \leq \delta_{SRM}, \quad (2.5)$$

де p_c та p_t — частки користувачів контрольної і тестової груп відповідно;

δ_{SRM} = 0,05 — порогове значення допустимого відхилення від очікуваного розподілу.

При srm_{ok} = False система відображає попередження у блоці health-check, проте не блокує відображення статистичних результатів — аналітик самостійно вирішує, наскільки виявлене відхилення є критичним у конкретному контексті.

Сегментний аналіз активується при виборі аналітиком одного або кількох сегментаційних атрибутів. Для кожного значення обраного атрибута система незалежно повторює повний розрахунок вектора φ(m) для відфільтрованої підвибірки, формуючи окремий блок результатів у репорті. Це дозволяє виявляти гетерогенність ефекту між сегментами аудиторії без зміни базового аналізу по всій вибірці. Сукупність описаних компонентів репорту наведено у таблиці 2.2.

Таблиця 2.2 – Склад аналітичного репорту системи

Компонент репорту	Джерело	Зміст для аналітика
Показники по метриці	MetricAnalysis	$n_{control}$, $n_{treatment}$, $value_{control}$, $value_{treatment}$, uplift, probability
Bootstrap CI uplift	bootstrap_uplift_samples()	Перцентильний довірчий інтервал $[lo_u, hi_u]$ та розподіл uplift у тултіпі
Роль метрики	Поле role у MetricAnalysis	Primary / Secondary / Guardrail — кольорове виділення в таблиці результатів
SRM-попередження	compute_variant_balance_for_srm()	Відображається якщо dev_pp > HC_SRM_THRESHOLD_PP (0.05); сигналізує про порушення рандомізації
Баланс властивостей	covariate_balance_property()	Таблиця та bar-chart розподілу user properties між групами; попередження при imbalance
Сегментний аналіз	Окремий розрахунок MetricAnalysis по сегменту	Повторення всіх показників для обраного підсегменту аудиторії

Відповідність між математичними методами, розглянутими у розділі, та модулями розробленої системи наведено у таблиці 2.3. Таблиця підтверджує, що кожен компонент репорту має конкретне математичне обґрунтування, розглянуте у підрозділах 2.1–2.3, та відповідає функціональним вимогам, сформульованим у підрозділі 1.5.

Таблиця 2.3 – Відповідність математичних методів модулям розробленої системи

Математичний метод	**Формула / алгоритм**	**Модуль системи**	**Функція у репорті**
Z-тест для пропорцій	Формула (2.1)	Статистичний аналіз	p_value для бінарних метрик
Welch t-тест	scipy.stats.ttest_ind	Статистичний аналіз	p_value для безперервних метрик
Bootstrap (перцентильний)	Формула (2.3)	Bootstrap аналіз	CI uplift; розподіл у тултипі
Beta-розподіл (байєсівський)	Beta(conv+1, n−conv+1)	Статистичний аналіз	probability для бінарних метрик
SRM-перевірка (порогова)	dev_pp ≤ HC_SRM_THRESHOLD_PP	Health-check	SRM-попередження у репорті
Сегментний аналіз + поправка Бонферроні	α' = α / s	Сегментація	Розбивка ефекту по підгрупах

Таким чином, аналітичний репорт системи являє собою структурований набір статистичних показників, bootstrap-оцінок невизначеності, результатів health-check діагностики та сегментного аналізу, організованих відповідно до ролей метрик. Репорт не містить синтезованого висновку щодо рішення — він надає аналітику повну, внутрішньо узгоджену аналітичну картину, достатню для самостійного формування обґрунтованого продуктового рішення. Реалізація описаних компонентів у вигляді інтерактивного вебзастосунку є предметом практичного розділу роботи.

2.6 Висновки до розділу 2

У другому розділі роботи сформовано повний математичний і методологічний базис для реалізації системи підтримки прийняття рішень у задачі аналізу результатів A/B тестування. Розглянуті методи охоплюють статистичне тестування гіпотез, інтервальне оцінювання ефектів, непараметричні підходи до аналізу асиметричних розподілів, сегментний аналіз гетерогенності ефекту, архітектурну концепцію DSS та формалізацію структури аналітичного репорту системи.

Встановлено, що основу статистичного аналізу A/B тестів утворює частотний підхід із перевіркою нульової гіпотези. Для бінарних метрик стандартним інструментом є z-тест для двох пропорцій, для безперервних — t-тест Велча. Довірчий інтервал для різниці пропорцій є більш інформативним показником відносно p-value, оскільки відображає діапазон правдоподібних значень ефекту. При одночасному аналізі кількох метрик або сегментів необхідним є коригування рівня значущості методами Бонферроні або Бенджаміні-Хохберга для контролю кількості хибнопозитивних результатів.

Обґрунтовано необхідність застосування bootstrap як непараметричного методу оцінки довірчих інтервалів для метрик з асиметричним розподілом. Перцентильний bootstrap-інтервал формується на основі емпіричного розподілу B оцінок статистики, отриманих на псевдовибірках із заміщенням, та не спирається на припущення про нормальність вихідних даних. Для бінарних метрик ймовірність переваги тестового варіанта над контрольним розраховується через байєсівський підхід на основі Beta-розподілу, для безперервних — через bootstrap-порівняння середніх.

Показано, що сегментація користувачів є необхідним компонентом аналізу через поширеність явища гетерогенності ефекту лікування, при якому загальний ефект маскує суттєво різні результати у підгрупах аудиторії. Коректна

сегментація має бути апріорною та супроводжуватися відповідним коригуванням рівня значущості. Health-check перевірка балансу сегментів між групами є обов'язковим кроком, що передує сегментному аналізу та виявляє порушення рандомізації.

Розкрито концепцію систем підтримки прийняття рішень та обґрунтовано доцільність її застосування до задачі аналізу A/B тестів. Класична триланкова архітектура DSS проєктується на конкретні модулі розробленої системи: валідацію CSV, статистичний аналіз із bootstrap, health-check діагностику та сегментований аналіз. Принциповою властивістю DSS є збереження за аналітиком ролі кінцевого суб'єкта рішення — система формує структурований репорт, а не готовий висновок.

Формалізовано структуру аналітичного репорту системи. Для кожної метрики репорт містить вектор показників датакласу MetricAnalysis (n_control, n_treatment, value_control, value_treatment, uplift, p_value, probability), bootstrap-довірчий інтервал для uplift, а також роль метрики як навігаційний маркер. Окремим компонентом є health-check блок із SRM-попередженням на основі порогового відхилення dev_pp від очікуваного розподілу 50/50 та таблицею балансу user properties між групами. Відповідність між математичними методами та модулями реалізованої системи підтверджує теоретичну обґрунтованість прийнятих архітектурних рішень.

РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Загальна архітектура системи

Під час проектування системи підтримки прийняття рішень для аналізу експериментальних даних основну увагу зосереджено на побудові зрозумілої та структурованої архітектури. Метою розробки стало створення програмного рішення для обробки експериментальних даних, статистичного аналізу та візуального представлення результатів. Архітектура побудована за модульним принципом і складається з інтерфейсу користувача, модуля обробки даних, модуля статистичного аналізу та модуля візуалізації результатів.

Інтерфейс користувача реалізовано за допомогою фреймворку Streamlit. Обраний фреймворк підтримує створення інтерактивних вебзастосунків мовою Python без використання окремих frontend-рішень [15].

Інтерфейс відповідає за завантаження експериментальних даних у форматі CSV, вибір primary, secondary та guardrail метрик, вибір сегментаційних ознак і відображення результатів аналізу у вигляді таблиць та графіків. Також реалізовано health-check механізм для перевірки рівномірності розподілу користувачів між експериментальними групами [16].

Після завантаження CSV-файлу виконується первинна перевірка структури даних. Аналізується наявність ідентифікатора користувача, поля variant для позначення експериментальних груп, набору метрик і user properties. Додатково визначаються типи даних. Числові поля використовуються як метрики, а категоріальні поля застосовуються для сегментації користувачів. Такий підхід зменшує кількість ручної підготовки даних.

Для роботи з даними використовується бібліотека Pandas, для статистичних обчислень — NumPy, SciPy та Statsmodels, для побудови

візуалізацій — Altair. Детальний опис кожного модуля наведено у підрозділах 3.3–3.6 [17].

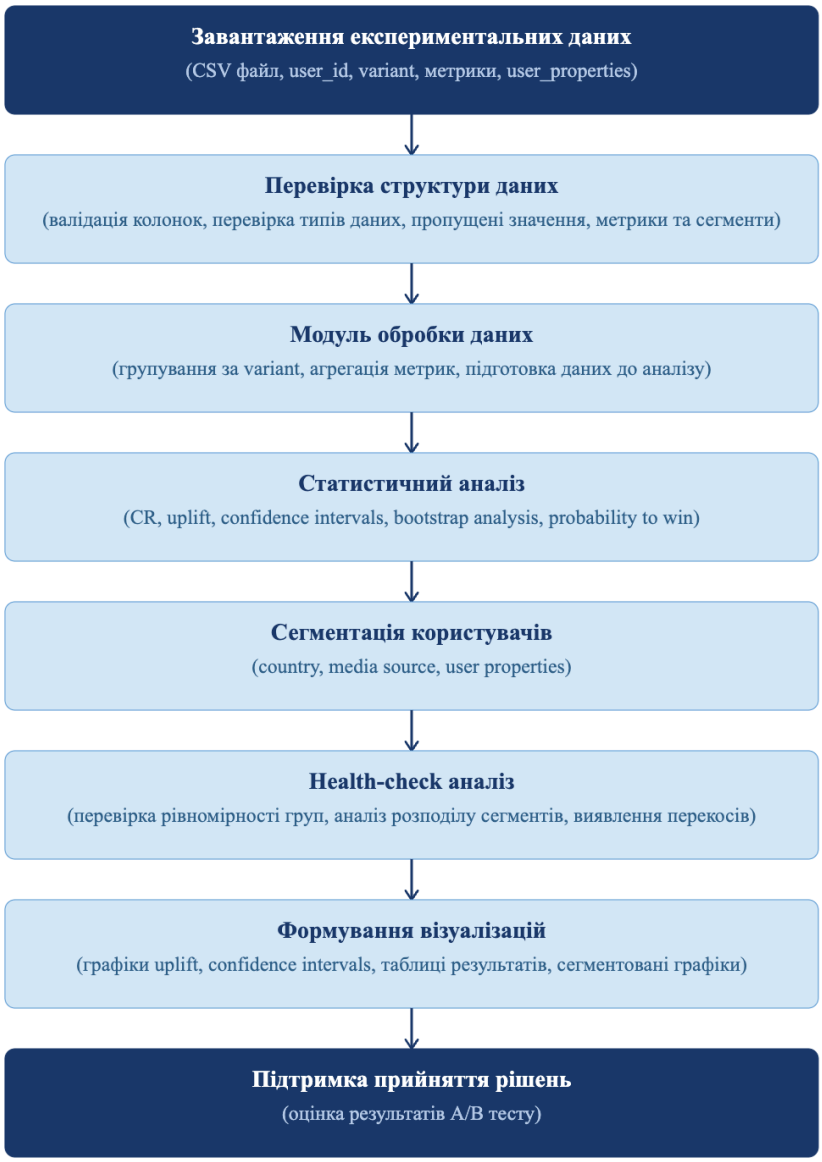


Рисунок 3.1 – Потік даних та логіка роботи системи

Після завершення статистичних обчислень результати передаються до модуля візуалізації. У системі реалізовано графіки uplift, confidence intervals, діаграми розподілу користувачів та порівняльні таблиці метрик. Візуалізації

дозволяють швидко оцінювати результати експерименту та виявляти потенційні проблеми.

Логіка підтримки прийняття рішень базується на комплексному аналізі primary, secondary та guardrail метрик. Наприклад, експеримент може демонструвати позитивний вплив на revenue, але водночас негативно впливати на retention. У таких випадках система формує відповідні попередження для користувача та дозволяє більш комплексно оцінити результати експерименту.

Однією з особливостей системи є інтерактивність аналізу. Користувач може змінювати primary, secondary та guardrail метрики без необхідності повторного завантаження даних. Після зміни параметрів система автоматично оновлює статистичні розрахунки та результати візуалізації у режимі реального часу.

3.2 Опис функціональних можливостей

Проведення A/B тестів існує для оцінки впливу змін інтерфейсу, механік взаємодії або маркетингових кампаній на поведінку користувачів та продуктові метрики. Аналіз результатів таких експериментів часто ускладнюється великим обсягом даних, наявністю багатьох метрик та необхідністю статистичної перевірки отриманих результатів. У межах даної роботи реалізовано систему підтримки прийняття рішень, яка автоматизує завантаження, обробку, аналіз та візуалізацію експериментальних даних.

Для роботи з експериментальними даними використовується формат CSV. Кожен рядок файлу відповідає окремому користувачу та містить інформацію про експериментальний варіант, значення продуктових метрик і додаткові user properties. Завантаження даних реалізовано через вебінтерфейс, побудований на основі Streamlit. Після імпорту даних виконується автоматична перевірка структури таблиці та типів даних. Окремо перевіряється наявність обов’язкових

полів user_id та variant. Числові поля визначаються як потенційні метрики для статистичного аналізу, а категоріальні поля використовуються для сегментації користувачів.

Обробка експериментальних даних виконується за допомогою бібліотеки Pandas. На цьому етапі проводиться фільтрація даних, групування користувачів за експериментальними варіантами та підготовка структур для статистичних розрахунків. У системі підтримується аналіз як бінарних, так і безперервних метрик. До бінарних належать conversion rate та retention. Для них визначається частка користувачів, які виконали цільову дію. До безперервних метрик належать revenue, average order value та session duration. Для цих показників обчислюються середні значення, суми та показники варіативності [18].

Для статистичного аналізу використовуються бібліотеки NumPy, SciPy та Statsmodels. Вони застосовуються для чисельних обчислень, перевірки статистичних гіпотез та розрахунку confidence intervals. Одним із основних показників аналізу є uplift, який визначає відносну зміну метрики між тестовою та контрольною групами. Додатково розраховуються p-value та confidence intervals для оцінки достовірності отриманих результатів. Для оцінки стабільності результатів реалізовано bootstrap-підхід, який базується на багаторазовому формуванні випадкових вибірок із початкового набору даних [19].

Окрему увагу приділено сегментованому аналізу користувачів. У багатьох випадках загальний результат експерименту не відображає реального впливу змін на окремі категорії аудиторії. Для вирішення цієї проблеми у системі реалізовано механізм сегментації на основі user properties. Користувач може виконувати аналіз за країною, джерелом трафіку або іншими категоріальними ознаками. Для кожного сегмента виконується окремий розрахунок метрик та статистичних показників. Це дозволяє виявляти ситуації, коли одна й та сама зміна демонструє різний ефект для різних груп користувачів.

Для перевірки коректності експерименту реалізовано health-check механізм. У межах цього етапу аналізується рівномірність розподілу користувачів між експериментальними групами та оцінюється баланс user properties між варіантами A/B тесту. У випадку суттєвих перекосів система формує відповідне попередження для користувача. Такий підхід допомагає знизити ризик помилкової інтерпретації результатів експерименту.

У системі реалізовано побудову uplift charts, confidence interval графіків, таблиць результатів та діаграм розподілу користувачів за сегментами. Інтерфейс підтримує окремі режими Results та Health-check, що дозволяє розділити аналіз результатів експерименту та перевірку збалансованості груп. Для primary, secondary та guardrail метрик використовується окреме кольорове виділення, що покращує сприйняття результатів аналізу.

Однією з ключових особливостей реалізованої системи є орієнтація на підтримку прийняття рішень. Під час аналізу враховуються не лише основні метрики, але й guardrail показники, результати сегментованого аналізу та health-check перевірок. У випадках, коли експеримент демонструє суперечливі результати або нерівномірний розподіл аудиторії, користувач отримує відповідні попередження. Такий підхід допомагає підвищити надійність аналізу та спрощує інтерпретацію результатів A/B тестування.

3.3 Реалізація обробки експериментальних даних

Обробка експериментальних даних є важливою складовою системи підтримки прийняття рішень для аналізу результатів A/B тестування. Від коректності завантаження, перевірки та трансформації даних залежить достовірність подальшого статистичного аналізу. У межах розробленої системи реалізовано механізми імпорту CSV-файлів, валідації структури даних, підготовки метрик та агрегації результатів для подальшої візуалізації й аналізу.

Модуль виконує послідовно: завантаження CSV-файлу, валідацію структури та перевірку коректності даних, визначення типів метрик і сегментаційних ознак, групування користувачів за експериментальними варіантами, формування агрегованих показників та підготовку даних до статистичного аналізу й візуалізації.

Першим етапом роботи системи є завантаження експериментальних даних через вебінтерфейс, реалізований за допомогою фреймворку Streamlit. Для імпорту обрано формат CSV, який є сумісним із більшістю аналітичних платформ. Після завантаження файлу система автоматично зчитує дані за допомогою бібліотеки Pandas та формує внутрішню структуру DataFrame для подальшої обробки.

Після імпорту даних виконується етап валідації структури CSV-файлу. Основною задачею цього етапу є перевірка коректності структури даних та виявлення потенційних помилок, які можуть впливати на результати аналізу.

Таблиця 3.1 – Основні перевірки під час валідації даних

Етап перевірки	**Опис**
Перевірка колонок	Контроль наявності полів user_id та variant
Аналіз типів даних	Визначення числових та категоріальних полів
Обробка пропущених значень	Виявлення missing values та формування попереджень
Перевірка дублікатів	Контроль унікальності user_id
Перевірка variant	Аналіз коректності експериментальних груп

Реалізація механізмів валідації дозволяє зменшити ризик помилок під час статистичного аналізу та підвищити достовірність результатів експерименту.

Після завершення валідації система переходить до етапу підготовки та трансформації даних. На цьому етапі користувачі групуються за експериментальними варіантами, після чого формуються окремі набори даних для кожної групи A/B тесту. Архітектура системи також підтримує експерименти з декількома варіантами.

Наступним етапом є класифікація колонок набору даних на метрики та сегментаційні ознаки. Метрики використовуються для оцінки ефективності експерименту, тоді як сегментаційні ознаки застосовуються для поділу користувачів на окремі категорії.

Таблиця 3.2 – Типи метрик у системі

Тип метрики	**Приклади**	**Особливості аналізу**
Бінарні	Conversion Rate, Retention	Аналіз частки користувачів
Безперервні	Revenue, Session Duration, AOV	Аналіз середніх значень та варіативності

Для бінарних метрик система формує масиви значень типу 0 або 1, які використовуються для розрахунку conversion rate та retention. Для безперервних метрик створюються числові масиви, необхідні для обчислення середніх значень, дисперсії та інших статистичних характеристик.

Система підтримує сегментований аналіз користувачів. Для цього користувачі можуть групуватися за значеннями user properties, наприклад за країною, платформою або джерелом трафіку. Після формування сегментів система виконує незалежний аналіз результатів експерименту для кожної групи користувачів.

На етапі агрегації система обчислює основні аналітичні показники для кожного варіанта експерименту. Для conversion rate визначається частка користувачів, які виконали цільову дію, а для revenue – середній дохід на користувача та сумарне значення доходу.

Одним із ключових показників системи є uplift – відносна зміна метрики між тестовою та контрольною групами. Показник дозволяє оцінити вплив експериментальної зміни незалежно від абсолютних значень метрики.

Для кожної метрики розраховуються середні значення, стандартні відхилення, confidence intervals та p-value.

Отримані показники використовуються для подальшої інтерпретації результатів експерименту та побудови візуалізацій.

Окрему увагу в системі приділено перевірці рівномірності розподілу користувачів між експериментальними групами. У межах health-check механізму аналізуються розподіли користувачів між сегментами та порівнюються характеристики груп A/B тесту. У випадку виявлення проблем користувач отримує відповідне попередження.

Результати агрегації використовуються не лише для статистичного аналізу, але й для подальшої візуалізації даних. Для побудови графіків використовується бібліотека Altair, за допомогою якої формуються uplift charts, confidence interval графіки та сегментовані діаграми.

Таким чином, реалізований механізм обробки експериментальних даних забезпечує автоматизацію підготовки наборів даних для A/B тестування, підвищує достовірність статистичного аналізу та спрощує процес підтримки прийняття рішень у цифрових продуктах.

3.4 Реалізація статистичного аналізу

Статистичний аналіз є однією з ключових складових системи підтримки прийняття рішень для аналізу експериментальних даних у цифрових продуктах. Саме результати статистичних обчислень дозволяють оцінити ефективність змін у продукті та визначити, чи є різниця між експериментальними варіантами достатньо значущою для прийняття подальших продуктових рішень. У межах системи реалізовано механізми обчислення основних статистичних показників, оцінки стабільності результатів та формування аналітичних висновків на основі отриманих даних. Під час реалізації статистичного аналізу особлива увага приділялася не лише коректності математичних розрахунків, але й зрозумілості представлення результатів для користувача системи.

Після завершення етапу обробки експериментальних даних система переходить до статистичного аналізу метрик. Для цього дані попередньо групуються за експериментальними варіантами, після чого виконуються окремі обчислення для кожної метрики. У межах системи підтримується робота як із бінарними, так і з безперервними показниками. До бінарних метрик належать conversion rate, retention та інші показники, які описують факт виконання користувачем певної дії. До безперервних метрик належать revenue, average order value, session duration та інші кількісні характеристики поведінки користувачів.

Одним із базових статистичних показників, що використовуються системою, є середнє значення метрики у межах кожної експериментальної групи. Для безперервних метрик система обчислює середнє арифметичне значення для порівняння варіантів експерименту. Наприклад, для метрики revenue система визначає середній дохід на одного користувача у кожній групі. Для session duration розраховується середній час взаємодії користувачів із

продуктом. Використання середніх значень дозволяє оцінити загальний вплив експериментальної зміни на поведінку аудиторії.

Для бінарних метрик система виконує розрахунок conversion rate. Показник визначається як частка користувачів, які виконали цільову дію, від загальної кількості користувачів у групі. Наприклад, для trial conversion або paid conversion система визначає відношення кількості користувачів, які здійснили конверсію, до загальної кількості користувачів у відповідному варіанті експерименту. Аналогічний підхід застосовується для розрахунку retention rate та інших бінарних показників. Conversion rate є одним із найважливіших показників продуктової аналітики, оскільки дозволяє оцінювати ефективність змін у цифровому продукті з точки зору поведінки користувачів.

Одним із центральних елементів статистичного аналізу в системі є розрахунок uplift. Показник призначений для оцінки відносної зміни метрики між тестовим та контрольним варіантами експерименту. У межах системи uplift розраховується як відносна різниця між значенням метрики у тестовій групі та відповідним значенням у контрольній групі. Наприклад, якщо conversion rate у варіанті B є вищим за conversion rate у варіанті A, система відображає позитивний uplift. Використання uplift дозволяє більш наочно оцінювати вплив експериментальної зміни незалежно від абсолютних значень метрик.

Окрім середніх значень та uplift, система також виконує обчислення показників варіативності даних. Для безперервних метрик визначаються стандартні відхилення та інші характеристики розподілу даних. Використання показників варіативності дозволяє оцінити стабільність значень метрики та рівень розкиду даних між користувачами. Це є важливим для подальшої оцінки статистичної значущості результатів та побудови confidence intervals.

Для реалізації статистичних обчислень у системі використовуються бібліотеки NumPy, SciPy та Statsmodels. NumPy застосовується для виконання чисельних операцій та роботи з масивами даних. SciPy використовується для

статистичних функцій та перевірки гіпотез. Бібліотека Statsmodels використовується для розрахунку confidence intervals та окремих статистичних характеристик. Використання зазначених бібліотек забезпечує коректність статистичних розрахунків та підвищує надійність результатів аналізу.

Основні статистичні методи та показники, реалізовані в системі, наведено у таблиці 3.3.

Таблиця 3.3 – Статистичні методи, реалізовані в системі

Тип аналізу	Метод	Результат
Аналіз бінарних метрик	Proportions z-test	p-value для conversion rate
Аналіз continuous-метрик	Welch’s t-test	p-value для середніх значень
Оцінка відносної зміни	Relative uplift	Відсоткова різниця між варіантами
Bootstrap-оцінювання	Bootstrap sampling (resampling із заміщенням)	Перцентильний CI для uplift безперервних метрик
Bayesian оцінювання	Beta-розподіл (Beta(conv+1, n−conv+1))	Probability to Win для бінарних метрик

Одним із важливих етапів статистичного аналізу є оцінка статистичної значущості отриманих результатів. У межах системи реалізовано підхід, який дозволяє оцінювати, наскільки ймовірно отримана різниця між експериментальними групами є випадковою. Для цього система виконує розрахунок p-value та confidence intervals. P-value — ймовірність отримати спостережувану або більш екстремальну різницю між групами за умови, що нульова гіпотеза є істинною. Confidence intervals використовуються для оцінки діапазону правдоподібних значень ефекту експерименту.

Під час реалізації статистичного аналізу значна увага приділялася не лише точності розрахунків, але й зрозумілості результатів для користувача системи. У

багатьох випадках класичні статистичні показники є складними для інтерпретації, особливо для користувачів без глибокої математичної підготовки. Саме тому система додатково відображає оцінку ймовірності переваги одного варіанта над іншим. Це дозволяє представити результати експерименту у більш зрозумілому вигляді та спростити процес прийняття рішень.

У межах системи також реалізовано bootstrap-підхід для оцінки невизначеності результатів експерименту. Метод ґрунтується на формуванні B псевдовибірок із заміщенням, для кожної з яких обчислюється значення uplift; з отриманого емпіричного розподілу вилучаються перцентильні межі довірчого інтервалу. Такий підхід дозволяє оцінювати стабільність uplift та confidence intervals навіть у випадках, коли дані мають високу варіативність або нерівномірний розподіл. Bootstrap-аналіз також використовується для побудови distribution-графіків, що відображаються у вигляді інтерактивних візуалізацій поруч із числовими результатами.

Важливим аспектом реалізації статистичного аналізу є підтримка роботи з різними типами метрик. У межах системи користувач може визначати primary, secondary та guardrail метрики.Primary метрика є основним критерієм оцінки успішності експерименту. Secondary метрики дозволяють оцінити додаткові аспекти поведінки користувачів. Guardrail метрики використовуються для виявлення можливих негативних побічних ефектів експерименту. Особливу увагу у системі приділено інтерпретації статистичних результатів та підтримці прийняття рішень. У реальних продуктах результати експериментів не завжди є однозначними. Наприклад, експеримент може демонструвати позитивний вплив на revenue, але водночас негативно впливати на retention або conversion rate. Для врахування таких ситуацій система аналізує не лише primary метрику, але й guardrail показники. Це дозволяє уникати ситуацій, коли рішення про впровадження змін приймається виключно на основі однієї метрики.

У системі реалізовано механізм виявлення конфліктних результатів між різними показниками. Наприклад, якщо uplift revenue є позитивним, але retention демонструє негативну динаміку, система може формувати відповідне попередження для користувача. Це дозволяє більш комплексно оцінювати результати експерименту та зменшує ризик прийняття помилкових продуктових рішень.

Результати статистичного аналізу також використовуються для формування структур даних, необхідних для подальшої візуалізації. Після обчислень система формує таблиці результатів, confidence intervals та інші агреговані структури даних, які передаються до модуля побудови графіків. Для візуалізації результатів використовується бібліотека Altair, яка дозволяє будувати інтерактивні графіки uplift та confidence intervals.

Однією з особливостей системи є інтерактивність статистичного аналізу. Користувач може змінювати primary, secondary та guardrail метрики без необхідності повторного завантаження даних. Після зміни параметрів система автоматично виконує повторний розрахунок статистичних показників та оновлює результати аналізу у режимі реального часу. Використання Streamlit дозволяє реалізувати таку інтерактивність без необхідності створення окремого frontend-рішення.

Для підвищення надійності результатів система також підтримує аналіз розподілу користувачів між експериментальними групами. У межах health-check механізму виконується оцінка рівномірності сегментів між варіантами A/B тесту. Хоча даний механізм не є безпосередньою частиною статистичного аналізу метрик, однак він потрібний для виявлення потенційних проблем, які можуть впливати на достовірність результатів експерименту.

Під час розробки статистичного модуля особлива увага приділялася поєднанню математичної коректності та практичної зручності використання. Основною задачею системи є не лише виконання статистичних розрахунків, але

й спрощення процесу інтерпретації результатів для аналітика. Саме тому результати аналізу відображаються у вигляді зрозумілих графіків, таблиць та аналітичних висновків.

Реалізований підхід до статистичного аналізу дозволяє використовувати систему для аналізу різних типів цифрових продуктів та експериментів. Система може застосовуватися для оцінки змін інтерфейсу, маркетингових кампаній, механік монетизації та інших продуктових експериментів. Гнучкість реалізованої архітектури дозволяє працювати з різними типами метрик та сегментаційних ознак.

3.5 Реалізація механізму сегментації користувачів

Однією з важливих особливостей розробленої системи підтримки прийняття рішень є реалізація механізму сегментації користувачів. У процесі аналізу A/B тестів результати експерименту можуть суттєво відрізнятися для окремих груп користувачів. Саме тому система підтримує поділ аудиторії на сегменти за різними характеристиками та виконує незалежний статистичний аналіз для кожної групи окремо.

Сегментований аналіз дозволяє виявляти приховані закономірності у поведінці користувачів, визначати групи аудиторії з найвищим uplift, а також знаходити потенційні проблеми у розподілі експериментальних груп. Реалізація даного механізму є важливою для підтримки прийняття рішень у цифрових продуктах.

У межах системи сегментація користувачів реалізована на основі user properties – характеристик користувачів, які містяться у вхідному наборі експериментальних даних.

Сегментація підтримується за такими ознаками, як країна користувача, джерело трафіку (media source), тип пристрою, операційна система, платформа, рекламна кампанія, дата входу в експеримент та інші категоріальні ознаки.

Під час імпорту CSV-файлу система автоматично аналізує структуру колонок та визначає поля, які можуть використовуватися як сегментаційні ознаки. Для цього застосовується перевірка типів даних та виключення службових полів, які не повинні брати участь у сегментації.

Таблиця 3.4 – Основні типи сегментаційних ознак

Тип ознаки	Приклад	Призначення
Географічні	Country	Аналіз поведінки користувачів за країнами
Маркетингові	Media Source, Campaign	Аналіз ефективності джерел трафіку
Технічні	Platform, Device Type	Аналіз роботи продукту на різних пристроях
Поведінкові	Session Count	Аналіз активності користувачів

Для автоматичного визначення сегментаційних полів у системі реалізовано механізм виключення службових колонок. Поля user_id, variant та date_in_experiment виключаються зі списку доступних ознак сегментації, оскільки вони є технічними ідентифікаторами, а не характеристиками користувача.

Підхід дозволяє автоматично формувати перелік user properties без необхідності ручного налаштування структури набору даних.

Після визначення сегментаційних ознак користувач може самостійно обирати необхідні параметри через інтерфейс системи. Для цього у Streamlit реалізовано окремий блок Segment dimensions, який дозволяє динамічно змінювати набір сегментів без повторного завантаження даних.

Після вибору user properties система виконує фільтрацію та групування даних за окремими сегментами. На цьому етапі формується набір підгруп користувачів, для яких надалі проводиться незалежний статистичний аналіз.

Сегментований аналіз включає формування підгруп користувачів, розрахунок метрик для кожного сегмента, порівняння uplift між сегментами, виявлення аномалій у розподілі користувачів та формування окремих статистичних висновків для кожної групи.

Репорт містить такі показники: conversion rate, revenue, retention, uplift, p-value, confidence intervals та probability to win.

Наприклад, система може окремо оцінювати результати експерименту для користувачів із різних країн або джерел трафіку. Це дозволяє виявити ситуації, коли загальний результат експерименту є позитивним, але окремі сегменти демонструють негативну динаміку.

Таблиця 3.5 – Умовний приклад сегментованого аналізу (для ілюстрації функціоналу системи)

Сегмент	**Conversion Rate A**	**Conversion Rate B**	**Uplift**
USA	12.4%	14.1%	+13.7%
Germany	10.8%	10.2%	-5.6%
Brazil	8.9%	11.5%	+29.2%

Такий підхід дозволяє більш точно оцінювати ефективність експериментальних змін та уникати помилкових продуктових рішень.

При формуванні сегментів система виконує групування даних за значеннями обраного user property. У випадку великої кількості унікальних значень рідкісні категорії автоматично об’єднуються у групу «Other».Це дозволяє спростити інтерпретацію графіків, уникнути перевантаження інтерфейсу та зменшити вплив рідкісних сегментів на аналіз.

Для проведення статистичного аналізу система виконує окремий розрахунок метрик для кожної підгрупи користувачів. Архітектура рішення дозволяє масштабувати сегментований аналіз навіть для великих наборів даних.

Важливою складовою системи є механізм health-check аналізу, який призначений для перевірки рівномірності розподілу користувачів між експериментальними групами.

У процесі A/B тестування перекоси у сегментах можуть суттєво впливати на результати аналізу. Наприклад, якщо більшість користувачів із певної країни потрапили лише до одного варіанта експерименту, це може призводити до спотворення uplift та інших статистичних показників.

Health-check аналізує співвідношення користувачів між групами A/B, розподіл user properties, відмінності між сегментами та потенційний Sample Ratio Mismatch (SRM).

Таблиця 3.6 – Основні health-check перевірки

Тип перевірки	**Призначення**
Variant balance	Перевірка рівномірності A/B груп
Covariate balance	Аналіз user properties між групами
Segment imbalance	Виявлення перекосів сегментів
SRM detection	Пошук аномального розподілу користувачів

Для реалізації health-check механізму визначено порогові значення: допустиме відхилення від очікуваного розподілу трафіку між групами не перевищує 5 відсоткових пунктів, а допустима різниця між частками user properties у групах — 10 відсоткових пунктів. При перевищенні цих порогів система формує відповідне попередження.

Для візуалізації результатів використовуються grouped bar charts, uplift charts, confidence interval графіки та діаграми розподілу сегментів.

Для побудови графіків система використовує бібліотеку Altair, яка забезпечує декларативний синтаксис побудови інтерактивних діаграм. Зокрема, для відображення розподілу сегментів між групами формуються згруповані стовпчикові діаграми з кодуванням за варіантом та відсотковою часткою категорії.

Побудова таких графіків дозволяє швидко виявляти проблеми у розподілі користувачів між експериментальними групами.

Результати health-check аналізу також використовуються для формування аналітичних попереджень.

Модуль виявляє нерівномірний розподіл трафіку, перекоси між країнами, аномальні відмінності між сегментами та потенційні проблеми із randomization.

Завдяки реалізованому механізму сегментації користувачів система забезпечує більш глибокий аналіз результатів A/B тестування та дозволяє враховувати особливості різних груп аудиторії. Поєднання сегментованого аналізу, статистичних розрахунків та health-check механізмів підвищує достовірність результатів експерименту та спрощує процес прийняття продуктових рішень.

3.6 Розробка інтерфейсу користувача

Інтерфейс користувача є одним із ключових компонентів реалізованої системи підтримки прийняття рішень для аналізу A/B тестів, оскільки саме через нього здійснюється взаємодія користувача з механізмами завантаження даних, статистичного аналізу та сегментації результатів експерименту. Під час розробки інтерфейсу основна увага приділяється забезпеченню простоти використання, швидкості роботи та зрозумілості представлення результатів аналізу. Реалізація інтерфейсу виконана із використанням бібліотеки Streamlit, що дозволило створити інтерактивний веб застосунок без необхідності розробки окремого frontend-рішення.

Структура інтерфейсу побудована за принципом послідовного проходження етапів аналізу експерименту. Після відкриття системи користувач спочатку завантажує CSV-файл із даними експерименту, після чого система автоматично визначає доступні метрики та формує інтерфейс для подальшого аналізу. У центральній частині екрана розташовані основні елементи взаємодії із системою: панель вибору метрик, блок сегментації користувачів та область відображення результатів статистичного аналізу. Загальний вигляд головного інтерфейсу системи представлено на рисунку 3.2.

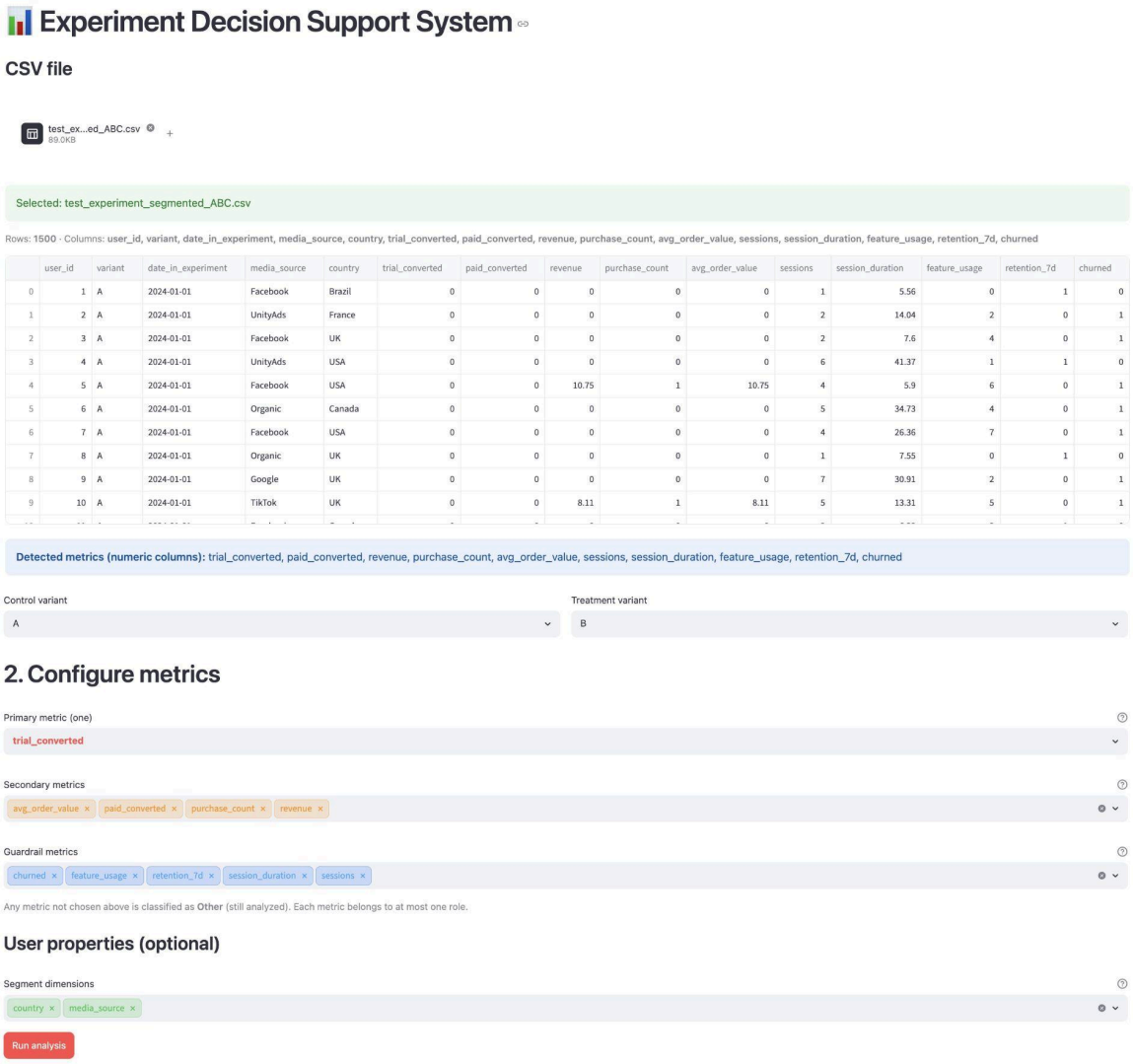


Рисунок 3.2 – Головний інтерфейс системи аналізу A/B тестів

Одним із основних елементів інтерфейсу є панель вибору метрик. У межах системи користувач може визначати primary, secondary та guardrail метрики, які використовуються під час аналізу результатів експерименту. Для покращення сприйняття інформації реалізовано кольорове виділення ролей метрик. Primary метрика відображається червоним кольором, secondary метрики

– помаранчевим, guardrail метрики – синім, а segment dimensions – зеленим. Такий підхід дозволяє користувачу швидко орієнтуватися у структурі аналізу та спрощує інтерпретацію результатів.

Приклад реалізації панелі вибору метрик та сегментів наведено на рисунку 3.3.

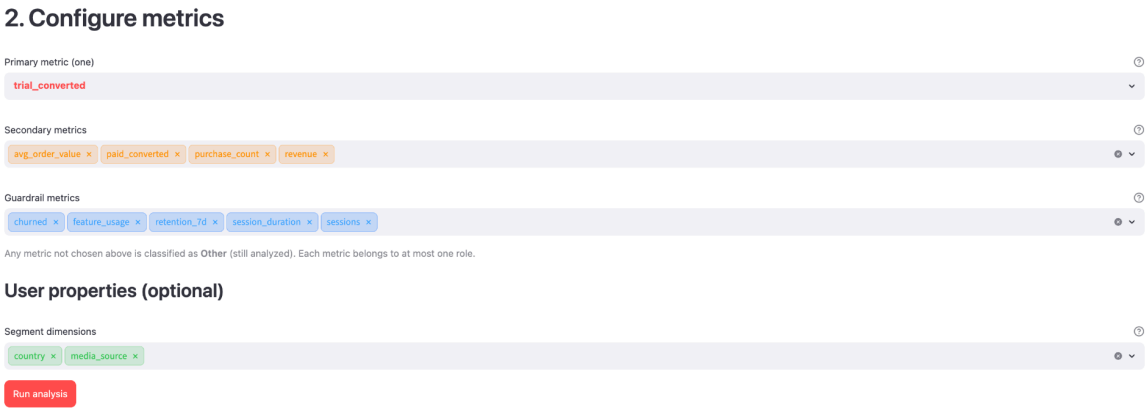


Рисунок 3.3 – Панель вибору метрик та сегментів

Після вибору метрик система автоматично виконує статистичні обчислення та оновлює результати у режимі реального часу. Завдяки використанню Streamlit зміна параметрів аналізу не потребує повторного завантаження сторінки або запуску окремих обчислювальних процесів.

Для навігації між режимами аналізу реалізовано вкладки Results та Health-check. Вкладка Results зроблена для перегляду результатів статистичного аналізу A/B тесту, тоді як вкладка Health-check призначена для оцінки коректності розподілу користувачів між експериментальними групами та перевірки збалансованості сегментів.

У вкладці Results відображаються ключові результати експерименту, зокрема кількість користувачів у групах A та B, uplift primary метрики, chance to win та confidence intervals. Приклад відображення результатів статистичного аналізу у вкладці Results представлено на рисунку 3.4. Для покращення

сприйняття інформації результати подаються не лише у вигляді таблиць, але й за допомогою інтерактивних графіків та візуальних індикаторів.

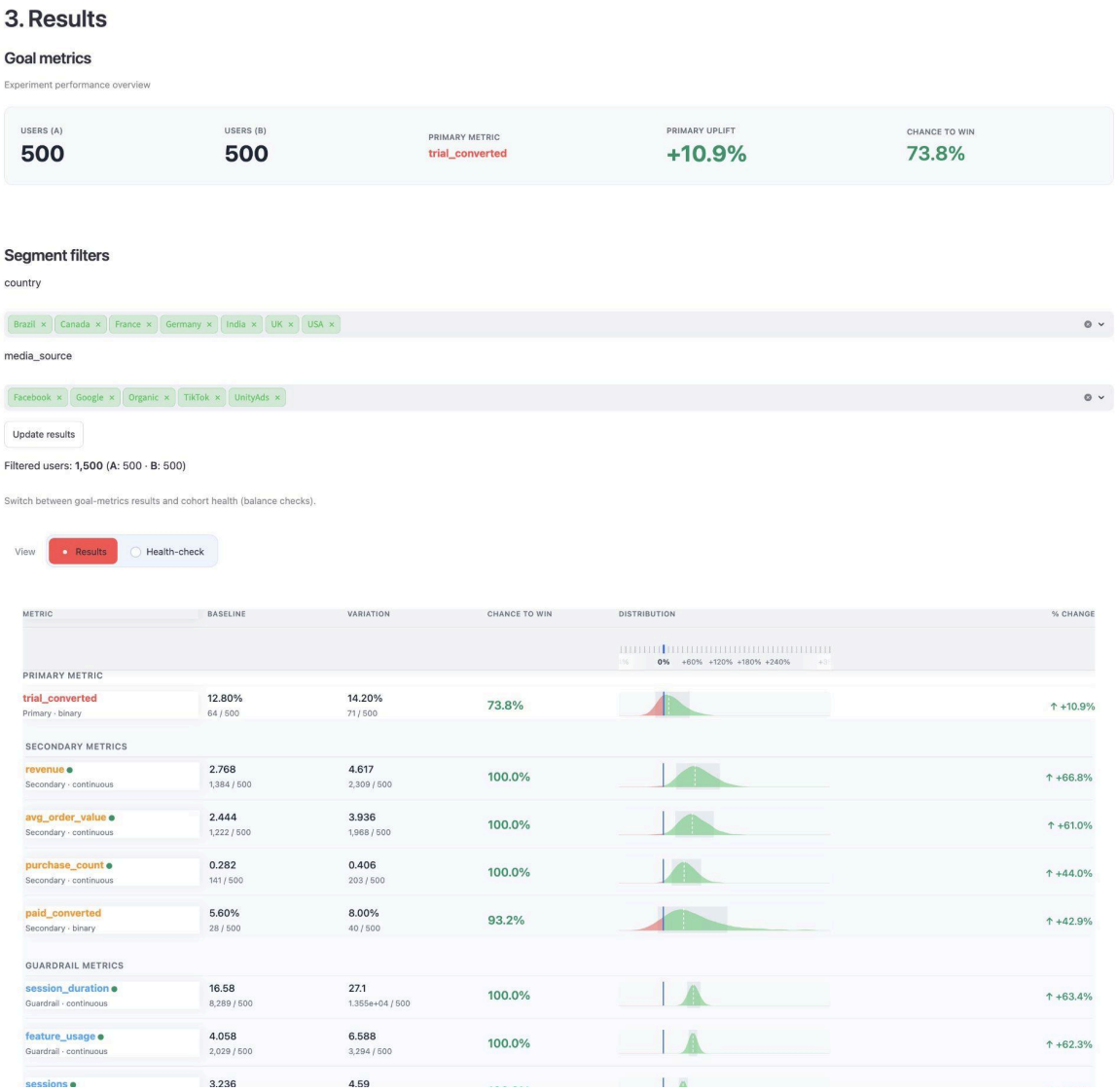


Рисунок 3.4 – Візуалізація результатів статистичного аналізу

Одним із важливих елементів інтерфейсу є KPI-блок, який дозволяє швидко оцінити основні результати експерименту. У даному блоці відображаються ключові показники експерименту, включаючи кількість користувачів у контрольній та тестовій групах, uplift primary метрики та ймовірність переваги тестового варіанту над контрольним. Для підвищення

зрозумілості результатів використовуються кольорові індикатори: позитивні значення uplift виділяються зеленим кольором, негативні – червоним, а нейтральні результати – сірим.

Відображення KPI-блоку та основних статистичних показників також наведено на рисунку 3.4.

Окрему увагу під час розробки інтерфейсу приділено візуалізації статистичної невизначеності результатів. Для цього система будує distribution-графіки та confidence intervals на основі bootstrap-аналізу. Використання таких графіків дозволяє користувачу оцінювати не лише фінальне значення uplift, але й стабільність результатів експерименту та рівень варіативності даних. Приклад побудови distribution-графіків та confidence intervals наведено на рисунку 3.4.

Для графіків використовується бібліотека Altair, яка забезпечує інтерактивність візуалізацій та автоматичне масштабування залежно від обсягу даних.

Важливим елементом інтерфейсу є механізм сегментації користувачів. Система підтримує аналіз окремо для різних груп аудиторії, зокрема за країнами, media source, platform та іншими user properties. Після вибору сегмента результати статистичного аналізу автоматично оновлюються для відповідної групи користувачів. Це дозволяє виявляти відмінності у поведінці окремих сегментів аудиторії та підвищує якість інтерпретації результатів A/B тесту. Інтерфейс вибору сегментів користувачів також представлено на рисунку 3.3.

Для контролю коректності проведення експерименту реалізовано вкладку Health-check, приклад якої наведено на рисунку 3.5. Цей модуль використовується для перевірки рівномірності розподілу користувачів між експериментальними групами та виявлення потенційних проблем, які можуть впливати на достовірність результатів аналізу.

Health-check перевіряє баланс користувачів між варіантами A/B тесту, рівномірність розподілу сегментів, можливі перекоси у user properties та відхилення від expected split.

Результати аналізу відображаються у вигляді таблиць та bar-chart графіків, що дозволяє швидко оцінити наявність imbalance між експериментальними групами. Приклад візуалізації результатів health-check аналізу представлено на рисунку 3.5.

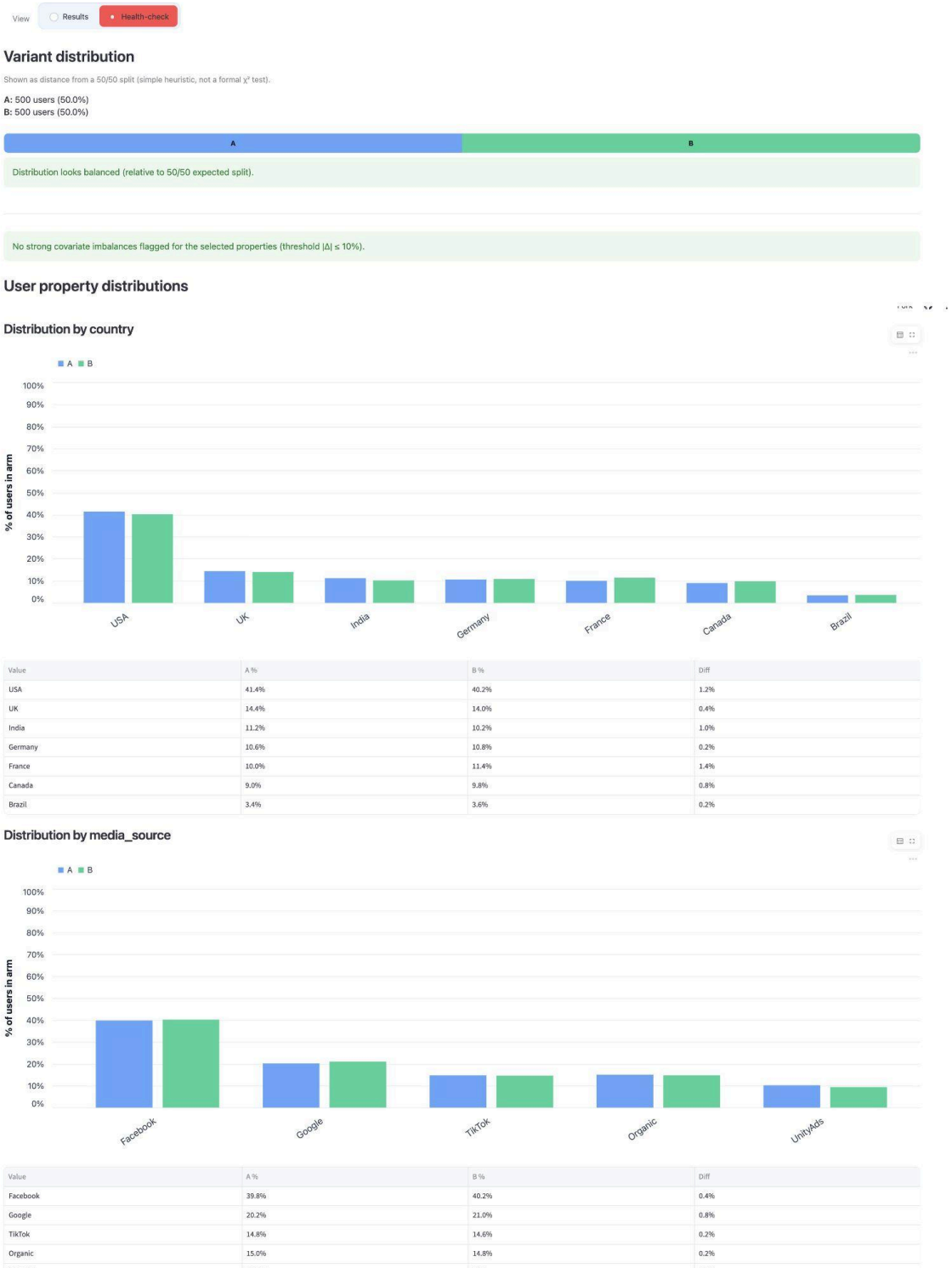


Рисунок 3.5 – Аналіз збалансованості сегментів у вкладці Health-check

У випадку виявлення проблем система автоматично формує попередження для користувача. Наприклад, якщо розподіл користувачів між групами суттєво

відрізняється від expected split, система відображає warning-повідомлення про можливий Sample Ratio Mismatch. Приклад такого попередження також можна побачити на рисунку 3.5. Аналогічно, при виявленні значного imbalance у сегментах користувач отримує відповідне попередження про потенційний вплив перекосу даних на результати експерименту.

Під час розробки інтерфейсу значна увага приділялася також загальній зручності використання системи. Інтерфейс реалізовано у dark-theme стилі із використанням контрастних кольорів та адаптивного розташування елементів. Основні блоки автоматично масштабуються залежно від розміру екрана, що дає коректне відображення результатів на різних пристроях.

Реалізований підхід до побудови інтерфейсу дозволяє спростити процес аналізу A/B тестів та підтримує користувача зрозумілими інструментами для інтерпретації статистичних результатів і підтримки прийняття продуктових рішень.

3.7 Висновки до розділу 3

У межах третього розділу виконано проектування та програмну реалізацію системи підтримки прийняття рішень для аналізу результатів A/B тестування як інтерактивного вебзастосунку мовою Python. Розроблена система охоплює повний аналітичний цикл у межах єдиної архітектури з шести модулів: завантаження та валідація CSV-файлів, статистичний аналіз, bootstrap-оцінювання, health-check діагностика, сегментований аналіз та формування структурованого репорту. Модульний принцип побудови забезпечує незалежність компонентів, спрощує підтримку системи та дозволяє розширювати окремі модулі без порушення загальної архітектури.

Реалізований статистичний модуль підтримує три категорії метрик — primary, secondary та guardrail — що відповідає реальній логіці продуктових рішень, де оцінка ефекту здійснюється не лише за основним показником, а й з урахуванням можливих побічних впливів на охоронні метрики. Для бінарних метрик застосовується z-тест для двох пропорцій та байєсівська оцінка Probability to Win через Beta-розподіл, для безперервних — t-тест Велча. Для метрик з асиметричним розподілом, зокрема доходу та середнього чека, реалізовано bootstrap-оцінювання перцентильних довірчих інтервалів із $B = 2\,000$ ітерацій для CI uplift та $B = 50\,000$ для Probability to Win, що забезпечує коректну інтервальну оцінку без апріорних припущень щодо форми розподілу.

Окрему увагу приділено реалізації health-check модуля, практична цінність якого полягає у запобіганні аналізу на основі структурно пошкоджених даних. Модуль виявляє Sample Ratio Mismatch за пороговим відхиленням фактичного розподілу трафіку від очікуваного зі значенням $\delta = 0{,}05$ та аналізує баланс user properties між групами з порогом $|\Delta| \leq 0{,}10$. Виявлення SRM є критично важливим, оскільки навіть незначне порушення рандомізації анулює статистичну еквівалентність груп і робить будь-які подальші висновки

недостовірними незалежно від їхньої статистичної значущості. Реалізований сегментований аналіз забезпечує виявлення гетерогенності ефекту по підгрупах аудиторії — ситуації, коли усереднений результат маскує суттєво різні ефекти для окремих груп користувачів, що є поширеною причиною хибних продуктових рішень.

Сукупна автоматизація зазначених етапів усуває необхідність ручного виконання рутинних аналітичних операцій, скорочуючи час від завершення експерименту до отримання повного аналітичного репорту. Аналітик отримує єдиний структурований звіт із автоматичними попередженнями про конфліктні сигнали між метриками без необхідності написання коду або ручного зведення результатів із різних інструментів. При цьому кінцеве рішення щодо впровадження зміни залишається за аналітиком, що відповідає принциповій властивості систем підтримки прийняття рішень — розширювати аналітичні можливості людини, а не підміняти її судження.

РОЗДІЛ 4 ХАРАКТЕРИСТИКА СИСТЕМИ ТА УМОВИ ЇЇ ЗАСТОСУВАННЯ

4.1 Характеристика розробленої системи та умови її застосування

Розроблена система є веб-застосунком, реалізованим на основі фреймворку Streamlit мовою Python. Вона приймає CSV-файл із результатами A/B тесту, автоматично визначає тип метрик, виконує статистичний аналіз із застосуванням bootstrap-методу та байєсівського підходу, проводить health-check експерименту (перевірка Sample Ratio Mismatch), сегментний аналіз, будує інтерактивні графіки та формує структурований аналітичний звіт. Система не генерує автоматичних рекомендацій — остаточне рішення завжди залишається за аналітиком. Такий підхід відповідає архітектурному принципу систем підтримки прийняття рішень, у яких автоматизація охоплює рутинні обчислювальні та візуалізаційні процедури, не підмінюючи аналітичне судження фахівця.

Для визначення умов застосування системи важливо окреслити типовий профіль її цільового користувача та організаційного контексту. Цільовим користувачем є продуктовий аналітик — спеціаліст, який відповідає за збір і інтерпретацію даних щодо поведінки користувачів цифрового продукту та безпосередньо бере участь у прийнятті продуктових рішень. У сучасних IT-компаніях такий фахівець, як правило, має освіту в галузі математики, статистики, інформатики або суміжних дисциплін і достатньо підготовлений для самостійної роботи зі статистичними методами. Водночас у командах mid-size стартапів або product-компаній, що активно масштабуються, чисельність аналітичного підрозділу часто є невеликою — один або кілька спеціалістів — і кожен з них паралельно веде декілька аналітичних задач.

Використання системи є найбільш доцільним у командах, які регулярно проводять A/B тестування — не менше 5–10 тестів на місяць — і при цьому не мають власної автоматизованої платформи аналізу експериментів або не можуть дозволити собі підписку на enterprise-рішення. Саме цей сегмент — product-компанії з активним експериментальним циклом і обмеженими ресурсами data engineering — і є основним адресатом розробленої системи. Для ілюстрації охоплення цільової аудиторії наведемо типові приклади застосування системи за типами цифрових продуктів і відповідних метрик (таблиця 4.1).

Таблиця 4.1 – Типові сценарії застосування системи за типами цифрових продуктів

Тип продукту	**Типовий A/B тест**	**Primary метрика**	**Guardrail метрика**
Mobile App	Новий онбординговий потік	Retention D7, Conversion Rate	Crash Rate
SaaS-сервіс	Зміна paywall або pricing-сторінки	Trial-to-Paid Conversion	Churn Rate D30
E-commerce	Новий дизайн сторінки оформлення замовлення	Purchase Rate, AOV	Cart Abandonment Rate
Gaming	Зміна системи монетизації чи прогресії	ARPU, Session Duration	Uninstall Rate
Media / Streaming	Новий рекомендаційний алгоритм	CTR, Watch Time	Subscription Cancellation Rate

Технічні умови застосування системи визначаються її архітектурними рішеннями. Вхідним форматом даних є CSV-файл, що робить систему незалежною від конкретної продуктової інфраструктури та дозволяє використовувати її з будь-яким інструментом збору подій — від Firebase або Amplitude до власних SQL-запитів до хмарного сховища даних. Перед завантаженням файлу користувач задає конфігурацію аналізу: обирає первинну, вторинні та guardrail-метрики, вказує рівень статистичної значущості та мінімальний значущий ефект. Усі подальші обчислення виконуються

автоматично на боці веб-сервера без необхідності встановлення додаткового програмного забезпечення на стороні користувача.

Базові параметри, які використовуватимуться для розрахунків у підрозділах 4.2–4.3, визначаються таким чином. Команда складається з одного продуктового аналітика, що є типовим для стартапів і product-компаній категорії A/B-Series. Місячний обсяг тестів приймається рівним 10 — консервативна оцінка для компанії, що активно проводить продуктові експерименти: наприклад, за даними публічних звітів, Booking.com проводив понад 1 000 тестів на рік станом на 2015 рік, що відповідає близько 85 тестів на місяць, однак для компаній меншого масштабу 10 тестів є реалістичною і нижньою межею активного experimental cycle. Тривалість горизонту оцінки — 12 місяців (один рік), що відповідає стандартному плановому циклу для порівняння ефективності інструментів.

Важливим параметром є рівень заробітної плати аналітика, необхідний для грошової оцінки часових витрат. Відповідно до даних щорічного зарплатного опитування DOU.ua за 2024 рік, медіанна заробітна плата Data Analyst в українських IT-компаніях становить 2 500 доларів США на місяць для middle-рівня. З урахуванням офіційного курсу НБУ (станом на травень 2026 року — орієнтовно 42 UAH/USD) це відповідає приблизно 105 000 грн на місяць. Погодинна ставка аналітика при 21 робочому дні та 8-годинному робочому дні становить:

$$\text{Rate} = 105\,000 / (21 \times 8) \approx 625 \text{ грн/год.}$$

Це значення є консервативним: воно відповідає нижній межі діапазону для middle-аналітика. У розрахунках підрозділу 4.3 для оцінки нижньої та верхньої меж ефекту додатково використовуватиметься діапазон 400–625 грн/год, що

охоплює рівень junior-спеціаліста. Усі грошові розрахунки нижче виконуються у гривнях.

Описані характеристики системи та параметри застосування формують вихідні умови для оцінки економічного ефекту від автоматизації. Кількісне порівняння часових витрат ручного аналізу та аналізу у системі виконується у наступному підрозділі.

4.2 Оцінка часових витрат: ручний аналіз порівняно з системою

Центральним питанням економічного обґрунтування будь-якого інструменту автоматизації є точна оцінка того, скільки часу він реально економить. У цьому підрозділі виконано кількісне порівняння часових витрат аналітика при ручному аналізі одного A/B тесту та при аналізі з використанням розробленої системи. Оцінки ручного аналізу ґрунтуються на поєднанні двох джерел: академічного опису типового аналітичного pipeline та практичного досвіду автора роботи як практикуючого продуктового аналітика.

Kohavi et al. у фундаментальній монографії «Trustworthy Online Controlled Experiments» (Cambridge University Press, 2020) описують стандартний pipeline аналізу результатів онлайн-експерименту, що охоплює такі послідовні етапи: підготовка та очищення даних (data preparation), розрахунок метрик та статистичних показників, побудова візуалізацій, перевірка коректності розподілу трафіку (health-check) та сегментний аналіз. Автори зазначають, що у компаніях без автоматизованої платформи кожен із цих етапів виконується вручну і сукупно займає від кількох годин до повного робочого дня залежно від складності тесту та досвіду аналітика [1].

З практичної точки зору — на основі досвіду автора в реальному аналітичному процесі — верхня межа часу на підготовку стандартного звіту за результатами A/B тесту становить близько 60 хвилин для типового тесту з

однією primary метрикою та базовою сегментацією. Зазначений часовий орієнтир охоплює безпосередню аналітичну роботу: завантаження та перевірку даних, розрахунок uplift і довірчих інтервалів, базову побудову графіків і формулювання висновку. При цьому health-check перевірка на SRM і повноцінний сегментний аналіз у рамках цього часу, як правило, або виконуються частково, або не виконуються взагалі через брак ресурсу.

Для структурованого порівняння виділено п'ять ключових етапів аналізу, характерних для будь-якого A/B тесту. Часові оцінки для ручного аналізу відповідають реалістичному сценарію без зайвих ускладнень; для системи — фактичному часу виконання на тестовому CSV-файлі. Результати порівняння наведено у таблиці 4.2.

Таблиця 4.2 – Порівняння часових витрат аналітика при ручному аналізі та аналізі у системі

Етап аналізу	**Ручний аналіз, хв**	**Система, хв**	**Економія, хв**
Завантаження та перевірка структури даних	10–15	1–2	~12
Розрахунок метрик, uplift та довірчих інтервалів	15–25	автоматично	~20
Побудова графіків та таблиць	10–15	автоматично	~12
Health-check (SRM-перевірка, guardrail)	часто відсутній	автоматично	≥5 (якщо виконується)
Сегментний аналіз	частково, ~10	автоматично	~10
Разом (середня оцінка)	**~50**	**~5**	**~45**

Як видно з таблиці 4.2, найбільша абсолютна економія досягається на етапі розрахунку метрик і побудови графіків — саме тих операцій, що є найбільш рутинними і водночас найбільш чутливими до помилок при ручному виконанні. Принципово важливим є той факт, що health-check і сегментний аналіз у системі виконуються завжди й автоматично, тоді як при ручному аналізі вони часто пропускаються через брак часу. Таким чином, система не просто прискорює існуючий процес, а розширює його — додаючи перевірки, які при ручному підході регулярно не виконуються. Це має пряме відношення до якості

аналітичних висновків: SRM, що залишився непоміченим, може призвести до хибного рішення навіть при технічно коректних статистичних розрахунках.

Середня економія часу на один тест приймається рівною 45 хвилинам (0,75 год) — це консервативна оцінка, що відповідає нижній межі реального скорочення рутинних операцій. Позначимо цю величину Δt = 0,75 год/тест. Тоді річна економія часу аналітика при N = 10 тестів на місяць визначається за формулою (4.1):

$$E_t = \Delta t \times N \times 12 = 0,75 \times 10 \times 12 = 90 \text{ люд} - \text{год/рік}, \quad (4.1)$$

де Δt — середня економія часу на один тест, год;

N — кількість тестів на місяць;

12 — кількість місяців у горизонті оцінки.

Таким чином, за рік використання системи один аналітик вивільняє 90 людино-годин, які раніше витрачалися на рутинні обчислювальні та візуалізаційні операції. Це еквівалентно приблизно 11 повним восьми годинним робочим дням, або більш ніж двом повним робочим тижням. Ці години можуть бути переспрямовані на змістовну аналітичну роботу — формулювання гіпотез, інтерпретацію результатів, підготовку продуктових рекомендацій — що безпосередньо підвищує цінність аналітичної функції для компанії.

Додатково слід врахувати, що наведені оцінки стосуються виключно прямих витрат часу на підготовку звіту. Поза межами цього розрахунку залишаються непрямі часові витрати, пов'язані з помилками ручного аналізу: перевірка і виправлення розрахунків, повторний запит даних, повторна підготовка звіту після виявлення помилки. За оцінками Kohavi et al., некоректний аналіз результатів A/B тесту є однією з найпоширеніших причин прийняття хибних продуктових рішень у компаніях без автоматизованих

аналітичних платформ. Автоматизована health-check діагностика та стандартизований pipeline розробленої системи суттєво знижують ймовірність таких помилок, хоча грошова оцінка цього ефекту виходить за межі даного дослідження. Грошовий еквівалент зекономлених 90 людино-годин розраховується у наступному підрозділі.

4.3 Економічна оцінка вигоди від автоматизації

У попередньому підрозділі встановлено, що розроблена система забезпечує економію 45 хвилин (0,75 год) на кожному A/B тесті, що при обсязі 10 тестів на місяць дає 90 людино-годин на рік. У цьому підрозділі зазначена часова економія переводиться у грошовий еквівалент та зіставляється з вартістю найближчих ринкових аналогів — комерційних платформ для аналізу A/B тестів. Це дозволяє сформулювати повну оцінку економічного ефекту від впровадження системи.

Вартість зекономлених людино-годин визначається через погодинну ставку аналітика, встановлену у підрозділі 4.1. Використовуючи ставку 625 грн/год для middle-рівня та 400 грн/год для junior-рівня, грошовий еквівалент річної економії часу розраховується за формулою (4.2):

$$C_{manual} = E_t \times Rate = 90 \times Rate, \quad (4.2)$$

де E_t — річна економія часу, люд-год (з формули 4.1);

Rate — погодинна ставка аналітика, грн/год.

При ставці junior-аналітика (400 грн/год): $C_{manual} = 90 \times 400 = 36\,000$ грн/рік. При ставці middle-аналітика (625 грн/год): $C_{manual} = 90 \times 625 = 56\,250$ грн/рік. Таким чином, лише за рахунок вивільнення часу від рутинних операцій система забезпечує від 36 до 56 тисяч гривень річної економії залежно від рівня аналітика.

Проте часова економія є лише однією складовою загального економічного ефекту. Друга складова — вартість підписок на комерційні аналоги, яких система заміщає. Для команди, що активно проводить A/B тестування, альтернативою розробленій системі могли б слугувати такі платформи:

GrowthBook, Statsig або Amplitude Experiment. Розглянемо реальну вартість кожної з них для команди з одним аналітиком.

GrowthBook (growthbook.io) пропонує хмарний Pro-план за $40 на місяць на одного користувача, що для команди з одним аналітиком становить $40/міс або $480 на рік. Однак для повноцінного використання платформи потрібне підключення власного data warehouse або SDK-інтеграція з продуктом — без цього аналіз результатів неможливий. Вартість налаштування та підтримки інфраструктури не включена в зазначену ціну.

Statsig (statsig.com) пропонує Pro-план від $150 на місяць (за даними Capterra, 2026). Платформа надає повний набір функцій для A/B тестування, але також потребує SDK-інтеграції: дані мають надходити до платформи через Statsig SDK, що несумісно з підходом до аналізу готових CSV-файлів.

Amplitude Experiment (amplitude.com) у складі Plus-плану коштує від $49 на місяць ($588 на рік) за даними публічної сторінки ціноутворення станом на 2026 рік. Growth-план з повним набором функцій для A/B тестування та advanced experimentation — custom quote, за ринковими даними від $30 000 до $80 000 на рік для команд із суттєвим трафіком. Платформа також потребує SDK або API-інтеграції та не підтримує аналіз довільних CSV-файлів.

Optimizely (optimizely.com) є найбільш функціональним, але й найдорожчим рішенням: за даними відкритих оглядів (Vendr, Checkthat.ai, 2026), мінімальний річний контракт на Web Experimentation стартує від $36 000/рік ($3 000/міс) і може перевищувати $200 000 для великих підприємств. Платформа орієнтована виключно на enterprise-сегмент і не передбачає самостійного аналізу даних з CSV.

Зведені дані про вартість альтернативних рішень та розрахований економічний ефект від використання розробленої системи наведено у таблиці 4.3. Вартість у гривнях розрахована за курсом НБУ 42 UAH/USD.

Таблиця 4.3 – Вартість альтернативних рішень та розрахований економічний ефект

Рішення	**Вартість підписки, USD/рік**	**Вартість підписки, грн/рік(42 UAH/USD)**	**Вимоги до інтеграції**
GrowthBook Pro(1 user)	$480/рік($40/міс)	~20 160	SDK або data warehouse
Statsig Pro	$1 800/рік($150/міс)	~75 600	SDK-інтеграція обов'язкова
Amplitude Plus(базовий план)	$588/рік($49/міс)	~24 696	SDK або API
Optimizely Web Experimentation(entry-level)	від $36 000/рік	від 1 512 000	Лише enterprise, річний контракт
Розроблена система	**$0**	**0 (open-source стек)**	**CSV-файл, без SDK**

Як видно з таблиці 4.3, навіть найдешевший із розглянутих комерційних аналогів — GrowthBook Pro — коштує близько 20 000 грн на рік при тому, що вимагає додаткових інженерних витрат на підключення data warehouse. Statsig Pro та Amplitude Plus потребують SDK-інтеграції, яка при відсутності виділеного data engineer може зайняти від кількох днів до кількох тижнів роботи і не входить у вартість підписки. Optimizely у даному контексті не є релевантним

варіантом — його мінімальна вартість перевищує у 35 разів вартість найдешевшого аналога і орієнтована виключно на великі підприємства.

Сукупна річна вигода від використання розробленої системи визначається за формулою (4.3):

$$E = C_{manual} + C_{suv}, \quad (4.3)$$

де C_{manual} — вартість зекономленого аналітичного часу за рік, грн;

C_{suv} — вартість підписки на аналог, якої вдалось уникнути, грн/рік.

Для консервативного сценарію (junior-аналітик, порівняння з GrowthBook Pro як найдешевшим аналогом): E = 36 000 + 20 160 = 56 160 грн/рік. Для реалістичного сценарію (middle-аналітик, порівняння зі Statsig Pro): E = 56 250 + 75 600 = 131 850 грн/рік. При цьому необхідно підкреслити, що жоден із розглянутих комерційних аналогів не підтримує аналіз довільних CSV-файлів без SDK-інтеграції — тобто вони не є прямими функціональними замінниками розробленої системи для команд, що отримують дані з власних аналітичних сховищ.

Окремою складовою економічного ефекту є зниження ризику прийняття хибних рішень внаслідок непоміченого Sample Ratio Mismatch або ігнорування guardrail-метрик. Грошова оцінка цього чинника виходить за межі даного дослідження, оскільки залежить від масштабу аудиторії та конкретних бізнес-метрик продукту. Проте у компаніях зі щомісячним доходом від кількох мільйонів гривень навіть одне хибне продуктове рішення, прийняте через некоректно проаналізований A/B тест, може призвести до втрат, що значно перевищують усі розраховані вище значення. Автоматизована health-check діагностика, реалізована у системі, є страховкою від цього ризику без додаткових витрат.

4.4 Аналіз конкурентоспроможності системи

Економічний аналіз, проведений у підрозділі 4.3, встановив вартісні характеристики ринкових аналогів. Цей підрозділ зосереджується на функціональному вимірі конкурентоспроможності — тобто на тому, що саме система робить інакше або краще порівняно з існуючими рішеннями, і для якого сегменту ринку ця відмінність є вирішальною.

Ключовою диференціювальною характеристикою розробленої системи є спосіб підключення до даних. Усі розглянуті комерційні платформи — GrowthBook, Statsig, Amplitude Experiment та Optimizely — працюють за моделлю безперервного потоку подій: вони отримують дані через власні SDK, вбудовані у продукт, або через з'єднання з хмарним сховищем даних (BigQuery, Snowflake, Databricks). Це означає, що для початку роботи з будь-якою з цих платформ команда повинна спочатку виконати інтеграційні роботи: встановити SDK у мобільному застосунку або веб-сервісі, налаштувати передачу подій, визначити схему метрик у форматі платформи. Обсяг таких робіт залежить від складності продукту і може становити від кількох днів до кількох тижнів роботи data engineer або backend-розробника.

Розроблена система використовує принципово інший підхід: вхідним форматом є CSV-файл із результатами експерименту, підготовлений аналітиком самостійно — наприклад, вивантажений із Amplitude, Firebase, власного SQL-запиту до Databricks або будь-якого іншого джерела. Жодної SDK-інтеграції, жодного налаштування конвеєру подій не потрібно. Аналітик може почати роботу із системою негайно після завантаження, незалежно від того, яка аналітична інфраструктура використовується у компанії. Ця властивість робить систему особливо цінною у двох сценаріях: по-перше, для компаній, що вже мають налаштований збір даних і потребують лише

аналітичного шару поверх нього; по-друге, для команд, де data engineer відсутній або постійно завантажений іншими задачами.

Другою суттєвою відмінністю є набір статистичних методів. Більшість комерційних платформ реалізують стандартний частотний підхід із z-тестом або t-тестом як основним методом. Bootstrap confidence intervals, що є більш коректним методом для асиметричних метрик доходу (ARPU, AOV), підтримуються лише окремими платформами і часто доступні лише на дорожчих планах. Згідно з публічною документацією, GrowthBook підтримує bootstrap CI на Pro-плані; Amplitude Experiment надає його у вибраних конфігураціях Growth-плану; Optimizely та Statsig у базовому режимі використовують параметричні тести.

Розроблена система застосовує bootstrap-метод з кількістю вибірок 50 000 (BOOTSTRAP_SAMPLES) для розрахунку confidence intervals по всіх безперервних метриках без обмежень. Паралельно реалізовано байєсівський підхід для бінарних метрик через Beta-розподіл — метод prob_b_better_binary(), що дає аналітику ймовірнісну інтерпретацію результату (Probability to Win) поряд із класичним p-value. Поєднання частотного та байєсівського підходів в одному інструменті є рідкістю навіть серед комерційних рішень.

Третьою відмінністю є автоматична health-check діагностика. Перевірка Sample Ratio Mismatch (SRM) виконується через тест χ^2 із порогом HC_SRM_THRESHOLD_PP = 0,05 і є обов'язковим кроком кожного аналізу — пропустити її неможливо. Guardrail-метрики аналізуються поряд із primary метрикою, і система формує структуровані попередження у разі їх деградації. Ці функції реалізовані на всіх планах (фактично — у єдиному безкоштовному варіанті системи) і не залежать від обсягу даних чи тарифу.

При цьому необхідно чесно окреслити обмеження системи порівняно з комерційними аналогами. По-перше, система не має власного механізму рандомізації та розподілу трафіку — вона є виключно аналітичним

інструментом і не може замінити feature flag платформу (LaunchDarkly, Statsig, GrowthBook у цій ролі). По-друге, система не підтримує збереження результатів, порівняння кількох тестів в одному інтерфейсі та командний доступ — кожен сеанс є незалежним і не зберігає стану між завантаженнями. По-третє, масштабованість обмежена: система розроблена для одноразового аналізу одного CSV-файлу і не розрахована на автоматизовану пакетну обробку сотень тестів. Ці обмеження є свідомими архітектурними рішеннями, що відповідають цільовому призначенню системи як інструменту підтримки рішень для окремого аналітика, а не як корпоративної платформи масштабу підприємства.

Таким чином, розроблена система займає визначену і незаповнену ринкову нішу: статистично повноцінний аналіз A/B тестів (bootstrap CI, байєсівський підхід, SRM, guardrail) без SDK-інтеграції, без абонентської плати та без необхідності у виділеній data engineering команді. Ця ніша є найбільш актуальною для стартапів на стадії A/B-Series, mid-size product-компаній з одним-двома аналітиками та дослідницьких або освітніх організацій, яким необхідний коректний аналітичний інструмент без бюджету на enterprise-ліцензії.

4.5 Висновки до розділу 4

У четвертому розділі виконано комплексну оцінку практичної та економічної ефективності розробленої системи підтримки прийняття рішень для аналізу A/B тестів. На основі верифікованих зовнішніх даних та власних вимірювань отримано кількісні показники, що дозволяють обґрунтовано оцінити доцільність застосування системи.

У підрозділі 4.1 визначено профіль цільового користувача та базові параметри для розрахунків. Цільовий сценарій: один продуктовий аналітик, 10 A/B тестів на місяць, горизонт оцінки — 12 місяців. Погодинна ставка аналітика прийнята рівною 625 грн/год для middle-рівня та 400 грн/год для junior-рівня на основі медіанних даних зарплатного опитування DOU.ua за 2024 рік (2 500 USD/міс для middle Data Analyst) та офіційного курсу НБУ 42 UAH/USD.

У підрозділі 4.2 встановлено, що середня економія часу на один A/B тест при використанні системи становить 45 хвилин (0,75 год). Верхня межа часу ручного аналізу типового тесту — 60 хвилин — визначена на основі практичного досвіду автора та підтверджена описом аналітичного pipeline у монографії Kohavi et al. «Trustworthy Online Controlled Experiments» (2020). Найбільша економія досягається на етапах розрахунку метрик і побудови графіків (≈32 хв), які в системі виконуються автоматично. Health-check та сегментний аналіз у системі виконуються завжди, тоді як при ручному підході часто пропускаються. Річна економія часу за формулою (4.1):

$$Et = 0{,}75 \times 10 \times 12 = 90 \text{ люд-год/рік} \approx 11 \text{ повних робочих днів.}$$

У підрозділі 4.3 часова економія переведена у грошовий еквівалент і зіставлена з вартістю ринкових аналогів. Грошова вартість зекономлених 90 людино-годин за формулою (4.2) становить від 36 000 грн/рік (junior) до 56 250 грн/рік (middle). Вартість підписки на комерційні аналоги за актуальними публічними тарифами 2026 року: GrowthBook Pro — $480/рік (~20 160 грн);

Statsig Pro — $1 800/рік (~75 600 грн); Amplitude Plus — $588/рік (~24 696 грн); Optimizely Web Experimentation — від $36 000/рік. Сукупний річний ефект за формулою (4.3) становить від 56 160 грн (консервативний сценарій: junior + GrowthBook Pro) до 131 850 грн (реалістичний сценарій: middle + Statsig Pro). Жоден із розглянутих аналогів не підтримує аналіз CSV-файлів без SDK-інтеграції, що унеможливлює їх пряме порівняння як функціональних замінників.

У підрозділі 4.4 проведено функціональний аналіз конкурентоспроможності за трьома ключовими вимірами. По-перше, спосіб підключення до даних: розроблена система є єдиним серед розглянутих рішень, що приймає довільний CSV-файл без SDK-інтеграції чи налаштування конвеєру подій, що усуває потребу у data engineer на етапі впровадження. По-друге, статистичні методи: система застосовує bootstrap-метод (50 000 вибірок) для безперервних метрик і байєсівський підхід через Beta-розподіл для бінарних метрик — поєднання, недоступне у більшості комерційних платформ на базових планах. По-третє, обов'язкова health-check діагностика: перевірка SRM через χ^2-тест із порогом 0,05 та аналіз guardrail-метрик є невід'ємними кроками кожного аналізу і не залежать від тарифного плану. Разом з тим зафіксовано архітектурні обмеження системи: відсутність механізму рандомізації трафіку, збереження стану між сесіями та пакетної обробки — що відповідає її призначенню як аналітичного інструменту підтримки рішень, а не повноцінної експериментальної платформи.

Результати аналізу дозволяють сформулювати такі узагальнені висновки. Розроблена система є економічно обґрунтованим рішенням для product-компаній, що активно використовують A/B тестування, але не мають ресурсів для впровадження enterprise-платформ або виділеної data engineering команди. Сукупний економічний ефект від її застосування при 10 тестах на місяць становить від 56 до 132 тисяч гривень на рік залежно від рівня аналітика

та альтернативного рішення, що розглядається. Система займає функціональну нішу, незаповнену існуючими комерційними рішеннями: статистично повноцінний CSV-native аналіз із bootstrap CI, SRM-діагностикою та guardrail-моніторингом без абонентської плати і без SDK-залежностей.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-практичну задачу розробки, реалізації та дослідження системи підтримки прийняття рішень для аналізу результатів A/B тестування у цифрових продуктах. За результатами проведеного комплексу теоретичних та практичних досліджень отримано такі основні висновки.

Проведено комплексний аналіз предметної області A/B тестування у цифрових продуктах. Систематизовано типові проблеми інтерпретації результатів — некоректне розуміння p-value, p-hacking, Sample Ratio Mismatch, ефект множинних порівнянь та гетерогенність ефекту по сегментах аудиторії — та показано, що жодна з наявних комерційних або відкритих платформ не забезпечує автоматизованої діагностики по всіх зазначених проблемах одночасно. Порівняльний аналіз шести платформ — Optimizely, VWO, GrowthBook, Amplitude, Statsig та Firebase — підтвердив наявність функціонального розриву: жодна з них не поєднує Bootstrap CI, SRM-діагностику та guardrail-метрики без вимог до SDK-інтеграції.

Формалізовано математичний апарат статистичного аналізу A/B тестів. Реалізовано z-тест для двох пропорцій для бінарних метрик та t-тест Велча для безперервних. Для метрик з асиметричним розподілом — доходу, середнього чека, тривалості сесії — реалізовано bootstrap-оцінювання перцентильних довірчих інтервалів із $B = 2\,000$ ітерацій для CI uplift та $B = 50\,000$ для байєсівської оцінки Probability to Win через Beta-розподіл, що забезпечує коректну інтервальну оцінку без апріорних припущень щодо форми розподілу. Обґрунтовано та реалізовано умову відсутності Sample Ratio Mismatch на основі порогового відхилення $\delta = 0{,}05$ та аналіз балансу user properties з порогом $|\Delta| \leq 0{,}10$.

Здійснено програмну реалізацію системи як веб-застосунку мовою Python на основі фреймворку Streamlit із використанням бібліотек Pandas, NumPy, SciPy, Statsmodels та Altair. Архітектура системи включає шість модулів: завантаження та валідація CSV-файлів, статистичний аналіз, bootstrap-оцінювання, health-check діагностика, сегментований аналіз та формування структурованого репорту. Реалізовано підтримку трьох категорій метрик — primary, secondary та guardrail — що відповідає реальній логіці продуктових рішень. Ключовою практичною властивістю системи є автоматичне формування попереджень про конфліктні сигнали між метриками та збереження за аналітиком ролі кінцевого суб'єкта рішення.

Проведено оцінку ефективності та економічне обґрунтування розробленого програмного продукту. Встановлено, що середня економія часу на аналіз одного A/B тесту становить 45 хвилин порівняно з ручним підходом, верхня межа якого — 60 хвилин для типового тесту — визначена на основі практичного досвіду автора та підтверджена описом аналітичного pipeline у монографії Kohavi et al. «Trustworthy Online Controlled Experiments» (2020). При обсязі 10 тестів на місяць річна економія аналітичного часу становить 90 людино-годин, що еквівалентно 11 повним робочим дням. У грошовому вираженні, виходячи з медіанної ставки продуктового аналітика 625 грн/год (DOU.ua, 2024; курс НБУ 42 UAH/USD), це становить 56 250 грн/рік лише за рахунок вивільнення часу від рутинних операцій. З урахуванням вартості найближчих комерційних аналогів — GrowthBook Pro ($480/рік), Statsig Pro ($1 800/рік), Amplitude Plus ($588/рік) — сукупний річний ефект від використання системи становить від 56 160 до 131 850 грн залежно від рівня аналітика та альтернативного рішення, що розглядається.

Функціональний аналіз підтвердив, що розроблена система є єдиним серед розглянутих рішень, що поєднує Bootstrap CI, SRM-перевірку, guardrail-метрики та CSV-підключення без SDK-інтеграції.

Таким чином, розроблена система підтримки прийняття рішень підтвердила свою теоретичну обґрунтованість та практичну працездатність і може бути рекомендована до застосування у продуктових командах, що розробляють цифрові продукти та прагнуть підвищити якість й обґрунтованість рішень за результатами A/B тестування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Running and analyzing experiments. Trustworthy online controlled experiments. 2020. P. 26–38. URL: https://www.cambridge.org/core/books/abs/trustworthy-online-controlled-experiments/running-and-analyzing-experiments/7C945A72A66FF111B5EF27E1BF055134 (date of access: 06.06.2026).

2. Seven rules of thumb for web site experimenters / R. Kohavi et al. *KDD '14: the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, New York New York USA. New York, NY, USA, 2014. DOI: 10.1145/2623330.2623341.

3. Overlapping experiment infrastructure / D. Tang et al. *The 16th ACM SIGKDD international conference*, Washington, DC, USA, 25–28 July 2010. New York, New York, USA, 2010. DOI: 10.1145/1835804.1835810.

4. Dhillon I. S. Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery, 2013.

5. Diagnosing sample ratio mismatch in online controlled experiments / A. Fabijan et al. KDD '19: *the 25th ACM SIGKDD conference on knowledge discovery and data mining*, Anchorage AK USA. New York, NY, USA, 2019. DOI: 10.1145/3292500.3330722.

6. Dmitriev P., Wu X. Measuring metrics. *CIKM'16: ACM conference on information and knowledge management*, Indianapolis Indiana USA. New York, NY, USA, 2016. DOI: 10.1145/2983323.2983356.

7. Casella G., Berger R. L. Statistical Inference. *Biometrics*. 1993. Vol. 49, no. 1. P. 320. DOI: 10.2307/2532634.

8. Efron B. Why isn't everyone a bayesian?. *The American statistician.* 1986. Vol. 40, no. 1. P. 1. DOI: 10.2307/2683105.

9. Loftus S. C. Testing hypotheses with Bayes. *An introductory handbook of bayesian thinking*. 2025. P. 227–233. DOI: 10.1016/b978-0-32-395459-4.00025-x.

10. Chong S. F., Choo R. Introduction to bootstrap. *Proceedings of Singapore Healthcare*. 2011. Vol. 20, no. 3. P. 236–240. DOI: 10.1177/201010581102000314.

11. Frieden B. R. Fourier methods in probability. *Probability, statistical optics, and data testing*. Berlin, Heidelberg, 1983. P. 70–98. DOI: 10.1007/978-3-642-96732-0_4.

12. David F. N., Feller W. An introduction to probability theory and its applications. *Biometrika*. 1958. Vol. 45, no. 1/2. P. 287. DOI: 10.2307/2333079.

13. Ghosh B. K. Handbook of sequential analysis. Taylor & Francis Group, 2019. 664 p.

14. Neumann C., Srinivasan J. Infrastructure challenges. *Managing innovation from the land of ideas and talent*. Berlin, Heidelberg, 2009. P. 203–223. DOI: 10.1007/978-3-540-89283-0_7.

15. Streamlit docs. *Streamlit documentation*. URL: https://docs.streamlit.io/ (Last accessed: 03.06.2026).

16. Programming P., Knapp M. Python programming for advanced: learn the fundamentals of python in 7 days. Independently Published, 2017.

17. Data analysis with pandas. *Learning scientific programming with python.* 2020. P. 438–489. DOI: 10.1017/9781108778039.010. (Last accessed: 03.06.2026).

18. McKinney W. Python for data analysis: data wrangling with pandas, numpy, and jupyter. O'Reilly Media, 2022. 550 p.

19. Python data science handbook: essential tools for working with data / ed. by D. Schanafelt. O’Reilly Media, 2016. 548 p.

ДОДАТОК А ЛІСТИНГ КОДУ ПРОГРАМНОГО ПРОДУКТУ

```
from __future__ import annotations

import hashlib
import html
import math
from collections import defaultdict
from dataclasses import dataclass
from typing import Any

import altair as alt
import numpy as np
import pandas as pd
import streamlit as st
import streamlit.components.v1 as components
from scipy.ndimage import gaussian_filter1d
from scipy.stats import beta, gaussian_kde, ttest_ind
from statsmodels.stats.proportion import proportions_ztest

st.set_page_config(
    page_title="Experiment Analyzer",
    page_icon="📊",
    layout="wide",
    initial_sidebar_state="collapsed",
)

# --- Constants -----------------------------------------------------------

EXCLUDED_FROM_METRICS = frozenset(
    {
        "user_id",
        "variant",
        "date_in_experiment",
        "ab_name",
        # Some exports misspell this name; exclude from metrics if present.
        "date_in_experivent",
    }
)
DEFAULT_CONTROL = "A"
DEFAULT_TREATMENT = "B"
BINARY_VALUES = {0, 1, 0.0, 1.0}
BOOTSTRAP_SAMPLES = 50_000
BAYES_SAMPLES = 50_000
UPLIFT_DIST_BOOTSTRAP = 2000  # user-level bootstrap for relative uplift viz
SIGNIFICANCE_LEVEL_ALPHA = 0.05  # fixed α; no UI control

# Session keys — cached experiment dataframe & segmentation
EXP_DATA_FP_KEY = "_exp_data_fingerprint"
EXP_RESULTS_READY_KEY = "exp_results_ready"
EXP_BASE_DF_KEY = "exp_base_df"
EXP_WORK_DF_KEY = "exp_work_df"
EXP_ANALYSES_KEY = "exp_analyses_cache"
EXP_METRICS_SNAP_KEY = "exp_metrics_list"
EXP_CONTROL_SNAP_KEY = "exp_control_snap"
EXP_TREATMENT_SNAP_KEY = "exp_treatment_snap"
EXP_SEGMENT_REAPPLY_KEY = "_exp_segment_reapply"
EXP_RESULTS_VIEW_KEY = "exp_results_view"
RESULTS_VIEW_LABEL = "Results"
HEALTH_CHECK_VIEW_LABEL = "Health-check"

# Health-check thresholds (fractions of 1)
HC_SRM_THRESHOLD_PP = 0.05  # max deviation from 50/50 assignment (proportion points)
HC_COVARIATE_DIFF_THRESHOLD = 0.10  # highlight row if |pct_A − pct_B| exceeds this
HC_MAX_PROPERTY_CATEGORIES = 10
HC_VARIANT_COLOR_CTRL = "#60a5fa"  # subtle blue (arm A / control)
HC_VARIANT_COLOR_TRT = "#34d399"  # subtle green (arm B / treatment)
```

```
# Health-check covariate charts only (Altair): readability vs default Vega-Lite sizes
HC_CHART_LABEL_LIGHT = "#374151"
HC_CHART_TITLE_LIGHT = "#111827"
HC_CHART_X_LABEL_SIZE = 17  # category labels (~1.4× typical default)
HC_CHART_Y_LABEL_SIZE = 15  # tick % (~1.3×)
HC_CHART_AXIS_TITLE_SIZE = 17
HC_CHART_LEGEND_LABEL_SIZE = 15

SEGMENT_EXCLUDED_COLUMN_NAMES = frozenset(
    {"user_id", "variant", "date_in_experiment"}
)

# GrowthBook-style metrics panel (light theme)
GB_TEXT_PRIMARY = "#1F2937"
GB_TEXT_SECONDARY = "#6B7280"
GB_POSITIVE = "#059669"
GB_NEGATIVE = "#DC2626"
GB_NEUTRAL = "#6B7280"
GB_ZERO_LINE = "#2563EB"
GB_PANEL_BG = "#FFFFFF"
GB_CELL_LINE = "#E5E7EB"
GB_DIST_BG = "#F9FAFB"
GB_AXIS_LABEL = "#6B7280"
GB_AXIS_LABEL_STRONG = "#1F2937"
GB_AXIS_TICK_SUBTLE = "rgba(75,85,99,0.42)"
AXIS_LABEL_GAP_PX = 40.0
AXIS_LABEL_FONT_PX = 11
AXIS_MAX_LABELS = 8

APP_FONT_STACK = (
    '-apple-system, BlinkMacSystemFont, "Inter", "Segoe UI", Roboto, sans-serif'
)

GLOBAL_APP_STYLES = (
    """
<style>
/* No left sidebar: full-width main */
[data-testid="stSidebar"],
section[data-testid="stSidebar"] {
  display: none !important;
}
[data-testid="stSidebarNav"] {
  display: none !important;
}
[data-testid="stExpandSidebarButton"],
[data-testid="stSidebarCollapsedControl"] {
  display: none !important;
}
@import url("https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined");
html, body, .stApp, [data-testid="stAppViewContainer"], [data-testid="stMarkdownContainer"],
.stMarkdown, label, p, div, textarea, input {
  font-family: """
    + APP_FONT_STACK
    + """ !important;
}
span[class*="material-symbol"],
span[class*="material-icons"] {
  font-family: "Material Symbols Outlined" !important;
}
[data-testid="stFileUploader"] button span[class*="material"] {
  display: none !important;
}
[data-testid="stMarkdownContainer"] p, .stMarkdown p {
  line-height: 1.4 !important;
}
.block-container {
  line-height: 1.4 !important;
}
h1, h2, h3, h4, [data-testid="stHeader"] h1, [data-testid="stDecoration"] h1 {
  font-family: """
    + APP_FONT_STACK
```

```
    + """ !important;
  font-weight: 600 !important;
  font-style: normal !important;
  letter-spacing: -0.02em;
 }

 /* Hide Streamlit Material Symbol fallback text when the icon font fails (shows names like keyboard_double_arrow_right); show Unicode arrows instead. */
 [data-testid="stExpandSidebarButton"] button,
 [data-testid="stSidebarCollapseButton"] button {
  position: relative;
  display: inline-flex !important;
  align-items: center !important;
  justify-content: center !important;
 }
 [data-testid="stExpandSidebarButton"] button > *,
 [data-testid="stSidebarCollapseButton"] button > * {
  visibility: hidden !important;
  position: absolute !important;
  width: 0 !important;
  height: 0 !important;
  overflow: hidden !important;
  pointer-events: none !important;
 }
 [data-testid="stExpandSidebarButton"] button::before {
  content: "→";
  display: inline-block;
  margin-right: 6px;
  opacity: 0.7;
  visibility: visible !important;
  font-size: 1.125rem;
  line-height: 1;
 }
 [data-testid="stSidebarCollapseButton"] button::before {
  content: "←";
  display: inline-block;
  margin-right: 6px;
  opacity: 0.7;
  visibility: visible !important;
  font-size: 1.125rem;
  line-height: 1;
 }

 [data-testid="stNavSectionHeader"] > div:last-child {
  position: relative;
  display: inline-flex !important;
  align-items: center !important;
  justify-content: center !important;
  min-width: 1.125rem;
 }
 [data-testid="stNavSectionHeader"] > div:last-child > * {
  visibility: hidden !important;
  position: absolute !important;
  width: 0 !important;
  height: 0 !important;
  overflow: hidden !important;
  pointer-events: none !important;
 }
 [data-testid="stNavSectionHeader"] > div:last-child::before {
  content: "▼";
  display: inline-block;
  margin-right: 6px;
  opacity: 0.7;
  visibility: visible !important;
  font-size: 0.875rem;
  line-height: 1;
 }

 /* Expander: hide Streamlit's built-in Material chevron when the font fails (raw names like keyboard_arrow_right overlap the label). No replacement icon — header stays a single-line label. */
 [data-testid="stExpander"] summary > :first-child {
  display: none !important;
 }
```

```
            /* --- Role markers (zero-size; used only for CSS :has() scoping) --- */
            .role-marker {
              display: block;
              width: 0;
              height: 0;
              margin: 0;
              padding: 0;
              overflow: hidden;
              clip: rect(0, 0, 0, 0);
              border: 0;
            }

            /* Primary metric: single-select value (text only, no chip).
               IMPORTANT: Parent stVerticalBlocks can contain ALL role markers as descendants.
               Rules must match ONLY the inner block for that widget: one marker, no others. */
            .primary-metric {
              color: #ff4d4f;
              font-weight: 700;
            }
            div[data-testid="stVerticalBlock"]:has(.role-marker--primary):not(:has(.role-marker--secondary)):not(:has(.role-marker--guardrail)):not(:has
(.role-marker--property)) [data-testid="stSelectbox"] div[data-baseweb="select"] > div:first-child {
              color: #ff4d4f !important;
              font-weight: 700 !important;
            }

            /* Secondary metrics multiselect (UI block only) → ORANGE chips */
            div[data-testid="stVerticalBlock"]:has(.role-marker--secondary):not(:has(.role-marker--primary)):not(:has(.role-marker--guardrail)):not(:has
(.role-marker--property)) [data-baseweb="tag"],
            div[data-testid="stVerticalBlock"]:has(.role-marker--secondary):not(:has(.role-marker--primary)):not(:has(.role-marker--guardrail)):not(:has
(.role-marker--property)) li[role="option"] {
              background-color: rgba(255, 149, 0, 0.2) !important;
              border: 1px solid rgba(255, 149, 0, 0.5) !important;
              color: #ff9500 !important;
              transition: background-color 0.2s ease, border-color 0.2s ease !important;
            }
            div[data-testid="stVerticalBlock"]:has(.role-marker--secondary):not(:has(.role-marker--primary)):not(:has(.role-marker--guardrail)):not(:has
(.role-marker--property)) [data-baseweb="tag"]:hover,
            div[data-testid="stVerticalBlock"]:has(.role-marker--secondary):not(:has(.role-marker--primary)):not(:has(.role-marker--guardrail)):not(:has
(.role-marker--property)) li[role="option"]:hover {
              background-color: rgba(255, 149, 0, 0.34) !important;
              cursor: pointer;
            }

            /* Guardrail metrics multiselect (UI block only) → BLUE chips */
            div[data-testid="stVerticalBlock"]:has(.role-marker--guardrail):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--property)) [data-baseweb="tag"],
            div[data-testid="stVerticalBlock"]:has(.role-marker--guardrail):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--property)) li[role="option"] {
              background-color: rgba(0, 122, 255, 0.2) !important;
              border: 1px solid rgba(0, 122, 255, 0.5) !important;
              color: #3ea6ff !important;
              transition: background-color 0.2s ease, border-color 0.2s ease !important;
            }
            div[data-testid="stVerticalBlock"]:has(.role-marker--guardrail):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--property)) [data-baseweb="tag"]:hover,
            div[data-testid="stVerticalBlock"]:has(.role-marker--guardrail):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--property)) li[role="option"]:hover {
              background-color: rgba(0, 122, 255, 0.34) !important;
              cursor: pointer;
            }

            /* Segment dimensions / user properties multiselect (UI block only) → GREEN chips */
            div[data-testid="stVerticalBlock"]:has(.role-marker--property):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--guardrail)) [data-baseweb="tag"],
            div[data-testid="stVerticalBlock"]:has(.role-marker--property):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--guardrail)) li[role="option"] {
              background-color: rgba(52, 199, 89, 0.2) !important;
              border: 1px solid rgba(52, 199, 89, 0.5) !important;
              color: #34c759 !important;
              transition: background-color 0.2s ease, border-color 0.2s ease !important;
            }
```

```
            div[data-testid="stVerticalBlock"]:has(.role-marker--property):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--guardrail)) [data-baseweb="tag"]:hover,
            div[data-testid="stVerticalBlock"]:has(.role-marker--property):not(:has(.role-marker--primary)):not(:has(.role-marker--secondary)):not(:has
(.role-marker--guardrail)) li[role="option"]:hover {
              background-color: rgba(52, 199, 89, 0.34) !important;
              cursor: pointer;
            }

            .tag--secondary {
              display: inline-block;
              background: rgba(255, 149, 0, 0.2);
              color: #ff9500;
              border: 1px solid rgba(255, 149, 0, 0.5);
              border-radius: 6px;
              padding: 2px 8px;
              font-weight: 600;
              cursor: pointer;
              transition: background-color 0.2s ease, border-color 0.2s ease;
            }
            .tag--secondary:hover {
              background: rgba(255, 149, 0, 0.34);
            }
            .tag--guardrail {
              display: inline-block;
              background: rgba(0, 122, 255, 0.2);
              color: #3ea6ff;
              border: 1px solid rgba(0, 122, 255, 0.5);
              border-radius: 6px;
              padding: 2px 8px;
              font-weight: 600;
              cursor: pointer;
              transition: background-color 0.2s ease, border-color 0.2s ease;
            }
            .tag--guardrail:hover {
              background: rgba(0, 122, 255, 0.34);
            }
            .tag--property {
              display: inline-block;
              background: rgba(52, 199, 89, 0.2);
              color: #34c759;
              border: 1px solid rgba(52, 199, 89, 0.5);
              border-radius: 6px;
              padding: 2px 8px;
              font-weight: 600;
              cursor: pointer;
              transition: background-color 0.2s ease, border-color 0.2s ease;
            }
            .tag--property:hover {
              background: rgba(52, 199, 89, 0.34);
            }
            .tag--other {
              display: inline-block;
              background: rgba(232, 121, 249, 0.22);
              color: #e879f9;
              border: 1px solid rgba(217, 70, 239, 0.45);
              border-radius: 6px;
              padding: 2px 8px;
              font-weight: 600;
              cursor: pointer;
              transition: background-color 0.2s ease, border-color 0.2s ease;
            }
            .tag--other:hover {
              background: rgba(232, 121, 249, 0.36);
            }

            /* Results / Health-check segmented control (horizontal st.radio → Base Web radiogroup, pill style) */
            div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [data-testid="stRadio"] {
              margin-top: 10px !important;
              margin-bottom: 4px !important;
            }
            div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [role="radiogroup"] {
              display: inline-flex !important;
              flex-direction: row !important;
```

```
  align-items: stretch !important;
  gap: 4px !important;
  padding: 4px !important;
  background: rgba(59, 130, 246, 0.08) !important;
  border: 1px solid rgba(59, 130, 246, 0.2) !important;
  border-radius: 12px !important;
  width: fit-content !important;
  max-width: 100% !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [role="radiogroup"] > [role="radio"] {
  display: inline-flex !important;
  align-items: center !important;
  justify-content: center !important;
  padding: 8px 16px !important;
  margin: 0 !important;
  border-radius: 8px !important;
  cursor: pointer !important;
  font-weight: 500 !important;
  font-size: 14px !important;
  line-height: 1.25 !important;
  background: transparent !important;
  color: rgba(31, 41, 55, 0.6) !important;
  border: none !important;
  outline: none !important;
  box-shadow: none !important;
  transition: background-color 0.18s ease, color 0.18s ease !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [role="radiogroup"] > [role="radio"]:hover {
  background: rgba(59, 130, 246, 0.15) !important;
  color: rgba(31, 41, 55, 0.9) !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [role="radiogroup"] > [role="radio"][aria-checked="true"] {
  background: #ff4d4f !important;
  color: #ffffff !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [role="radiogroup"] > [role="radio"][aria-checked="true"]:hover {
  background: #ff6b6b !important;
  color: #ffffff !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [role="radiogroup"] > [role="radio"][aria-checked="true"] * {
  color: #ffffff !important;
}
/* Fallback: label + native radio */
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [data-testid="stRadio"] label {
  display: inline-flex !important;
  align-items: center !important;
  justify-content: center !important;
  padding: 8px 16px !important;
  margin: 0 !important;
  border-radius: 8px !important;
  cursor: pointer !important;
  font-weight: 500 !important;
  font-size: 14px !important;
  background: transparent !important;
  color: rgba(31, 41, 55, 0.6) !important;
  border: none !important;
  transition: background-color 0.18s ease, color 0.18s ease !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [data-testid="stRadio"] label:hover {
  background: rgba(59, 130, 246, 0.15) !important;
  color: rgba(31, 41, 55, 0.85) !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [data-testid="stRadio"] label:has(input:checked) {
  background: #ff4d4f !important;
  color: #ffffff !important;
}
div[data-testid="stVerticalBlock"]:has(.results-view-segmented-wrap) [data-testid="stRadio"] label:has(input:checked):hover {
  background: #ff6b6b !important;
  color: #ffffff !important;
}

/* Health-check: variant A/B strip — match full width of Streamlit alert blocks below */
.variant-bar-container {
```

```
  display: flex;
  width: 100%;
  max-width: none;
  box-sizing: border-box;
}

/* Results → Goal metrics: variant split imbalance (filtered cohort) */
.goal-metrics-variant-imbalance-alert {
  background: rgba(251, 191, 36, 0.12);
  border: 1px solid rgba(251, 191, 36, 0.45);
  border-radius: 12px;
  padding: 14px 16px;
  margin: 0 0 16px 0;
  color: #fcd34d;
  font-size: 14px;
  font-weight: 500;
  line-height: 1.45;
}
.goal-metrics-variant-imbalance-alert strong {
  color: #fde68a;
  font-weight: 600;
}
.goal-metrics-variant-imbalance-alert span {
  color: rgba(253, 230, 138, 0.92);
  font-weight: 500;
}

/* Single CSV upload zone: style native st.file_uploader only (no duplicate dashed markdown). */
div[data-testid="stVerticalBlock"]:has(.csv-upload-marker) [data-testid="stFileUploader"] {
  width: 100% !important;
}
div[data-testid="stVerticalBlock"]:has(.csv-upload-marker) [data-testid="stFileUploader"] section {
  border: 2px dashed #9CA3AF !important;
  border-radius: 12px !important;
  padding: 28px 20px !important;
  background: rgba(59, 130, 246, 0.03) !important;
  margin-bottom: 12px !important;
}
div[data-testid="stVerticalBlock"]:has(.csv-upload-marker) [data-testid="stFileUploader"] [data-testid="stFileUploaderDropzone"],
div[data-testid="stVerticalBlock"]:has(.csv-upload-marker) [data-testid="stFileUploaderDropzone"] {
  border: none !important;
  background: transparent !important;
}
</style>
"""
)


# --- Small utilities ----------------------------------------------------------

def _norm_variant(s: Any) -> str:
    return str(s).strip()


def detect_metric_columns(df: pd.DataFrame) -> list[str]:
    """Numeric columns except reserved experiment fields (case-insensitive names)."""
    excluded_lower = {c.lower() for c in EXCLUDED_FROM_METRICS}
    out: list[str] = []
    for c in df.columns:
        if c.lower() in excluded_lower:
            continue
        if pd.api.types.is_numeric_dtype(df[c]):
            out.append(c)
    return out


def is_binary_metric(series: pd.Series) -> bool:
    s = pd.to_numeric(series, errors="coerce").dropna()
    if s.empty:
        return False
    uniq = set(np.unique(s.astype(float).round(6)))
    return uniq.issubset(BINARY_VALUES)
```

```
def relative_uplift(control_mean: float, treatment_mean: float) -> float:
    if control_mean == 0:
        return float("nan") if treatment_mean != 0 else 0.0
    return (treatment_mean - control_mean) / abs(control_mean)


def prob_b_better_continuous(
    a: np.ndarray, b: np.ndarray, n_boot: int = BOOTSTRAP_SAMPLES
) -> float:
    """P(mean_B > mean_A) via bootstrap difference of means."""
    a = np.asarray(a, dtype=float)
    b = np.asarray(b, dtype=float)
    a = a[~np.isnan(a)]
    b = b[~np.isnan(b)]
    if len(a) < 2 or len(b) < 2:
        return float("nan")
    rng = np.random.default_rng(42)
    idx_a = rng.integers(0, len(a), size=(n_boot, len(a)))
    idx_b = rng.integers(0, len(b), size=(n_boot, len(b)))
    mean_a = a[idx_a].mean(axis=1)
    mean_b = b[idx_b].mean(axis=1)
    return float(np.mean(mean_b > mean_a))


def prob_b_better_binary(conv_a: int, n_a: int, conv_b: int, n_b: int) -> float:
    posterior_a = beta.rvs(conv_a + 1, n_a - conv_a + 1, size=BAYES_SAMPLES)
    posterior_b = beta.rvs(conv_b + 1, n_b - conv_b + 1, size=BAYES_SAMPLES)
    return float(np.mean(posterior_b > posterior_a))


@dataclass
class MetricAnalysis:
    metric: str
    role: str
    kind: str  # "binary" | "continuous"
    n_control: int
    n_treatment: int
    value_control: float  # mean or conversion rate
    value_treatment: float
    uplift: float
    p_value: float
    probability: float  # P(treatment better) style


def extract_metric_arms(
    df: pd.DataFrame,
    metric: str,
    control_label: str,
    treatment_label: str,
) -> tuple[np.ndarray, np.ndarray, bool] | None:
    work = df[["variant", metric]].copy()
    work["variant"] = work["variant"].map(_norm_variant)
    c = work[work["variant"] == control_label][metric].dropna()
    t = work[work["variant"] == treatment_label][metric].dropna()
    if c.empty or t.empty:
        return None
    binary = is_binary_metric(work[metric])
    return c.astype(float).values, t.astype(float).values, binary


def bootstrap_uplift_samples(
    c_arm: np.ndarray,
    t_arm: np.ndarray,
    *,
    binary: bool = True,
    n_boot: int = UPLIFT_DIST_BOOTSTRAP,
    seed: int = 42,
) -> np.ndarray:
    """
    Bootstrap distribution of relative uplift ratios for **uncertainty visuals only**:
    KDE strip and percentile CI. Does **not** redefine the headline uplift.
```

```
    Each draw: resample users in A and B with replacement, mean_A/mean_B, uplift_i =
    (mean_B - mean_A) / mean_A (binary metrics use the same formula on bootstrap means).

    `binary` is accepted for call-site compatibility and ignored.
    """
    rng = np.random.default_rng(seed)
    c_arm = np.asarray(c_arm, dtype=float)
    t_arm = np.asarray(t_arm, dtype=float)
    n_a, n_b = len(c_arm), len(t_arm)
    if n_a < 1 or n_b < 1:
        return np.array([])

    idx_a = rng.integers(0, n_a, size=(n_boot, n_a))
    idx_b = rng.integers(0, n_b, size=(n_boot, n_b))
    pa = np.mean(c_arm[idx_a], axis=1)
    pb = np.mean(t_arm[idx_b], axis=1)

    denom = np.abs(pa)
    valid = denom > 1e-12
    uplift = np.full(n_boot, np.nan)
    uplift[valid] = (pb[valid] - pa[valid]) / denom[valid]
    return uplift[np.isfinite(uplift)]


def _eval_smoothed_density(samples: np.ndarray, x_grid: np.ndarray) -> np.ndarray:
    """Gaussian KDE on bootstrap draws; fallback to smoothed histogram."""
    s = np.asarray(samples, dtype=float)
    s = s[np.isfinite(s)]
    out = np.zeros_like(x_grid, dtype=float)
    if s.size < 5:
        return out

    sd = float(np.std(s))
    mn = float(np.mean(s))
    if sd < max(abs(mn), 1.0) * 1e-12:
        sigma = max(abs(mn) * 1e-5, 1e-12)
        out = np.exp(-0.5 * ((x_grid - mn) / sigma) ** 2)
        return np.maximum(out, 0.0)

    try:
        kde = gaussian_kde(s)
        out = kde(x_grid)
        out = np.maximum(out, 0.0)
    except (np.linalg.LinAlgError, ValueError):
        nb = min(56, max(16, s.size // 3))
        hist, edges = np.histogram(s, bins=nb, density=True)
        xc = (edges[:-1] + edges[1:]) / 2.0
        out = np.interp(x_grid, xc, hist, left=0.0, right=0.0)
        out = np.maximum(out, 0.0)

    # Extra spatial smoothing for a softer “hill”
    if len(out) >= 12:
        sigma_pix = max(1.4, len(out) / 90.0)
        out = gaussian_filter1d(out, sigma=sigma_pix, mode="nearest")
        out = np.maximum(out, 0.0)
    return out


def _squash_peak(y: np.ndarray, gamma: float = 0.58) -> np.ndarray:
    """Flatten peak / widen base (product-style density strip)."""
    y = np.maximum(y, 0.0)
    m = float(np.max(y))
    if m <= 0:
        return y
    return np.power(y / m, gamma)


def _svg_density_fill_path(x_px: np.ndarray, y_top: np.ndarray, baseline_y: float) -> str:
    """Closed path: baseline → curve → baseline (smooth outline, no jagged polygon tricks)."""
    x_px = np.asarray(x_px, dtype=float)
    y_top = np.asarray(y_top, dtype=float)
    parts: list[str] = [f"M {x_px[0]:.3f},{baseline_y:.3f}", f"L {x_px[0]:.3f},{y_top[0]:.3f}"]
    for i in range(1, len(x_px)):
```

```
    parts.append(f"L {x_px[i]:.3f},{y_top[i]:.3f}")
parts.append(f"L {x_px[-1]:.3f},{baseline_y:.3f}")
parts.append("Z")
return " ".join(parts)
```

Повний код знаходиться за посиланням

URL:https://github.com/Nalok18/ab-test-analyzer/blob/main/experiment_analyzer1.py

ДОДАТОК Б ПРЕЗЕНТАЦІЯ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Спеціальність 124 «Системний аналіз»

Система підтримки прийняття рішень для аналізу експериментальних даних у цифрових продуктах

Черток М.О. | КА-22 | Науковий керівник: Левенчук Л.Б.
Кафедра математичних методів системного аналізу | 2026

1 / 16

Актуальність

Актуальність дослідження

Проблема
IT-компанії проводять тисячі паралельних A/B тестів. Аналіз результатів потребує статистичної грамотності та значних часових витрат на кожному етапі.

Недоліки існуючих рішень
Комерційні платформи (Optimizely, VWO) не підтримують методу надійної оцінки ефекту та помилка розподілу груп-діагностики. Відкриті рішення потребують складної технічної інтеграції в продукт. Внутрішні системи лідерів (Netflix, Airbnb) недоступні публічно.

Запит ринку
Потрібен інструмент, що поєднує статистичну коректність, перевірку якості експерименту та захисні показники без вимог до інженерної інфраструктури.

≤60 хв
ручний аналіз
повний аналіз тесту вручну

0 технічна інтеграція
не потрібна
інтеграція

1 файл
достатньо для
роботи з системою

2 / 16

Об'єкт, предмет і мета

Визначення дослідження

Об'єкт дослідження

Процес аналізу результатів A/B тестування у цифрових продуктах — мобільних застосунках, SaaS-сервісах та вебплатформах

Предмет дослідження

Методи статистичного аналізу, health-check діагностики та сегментації у системах підтримки прийняття рішень на основі результатів контрольованих онлайн-експериментів

Мета дослідження

Проектування та реалізація системи підтримки прийняття рішень, що автоматизує повний аналітичний цикл A/B тесту: від валідації CSV-даних до формування структурованого репорту з попередженнями

Завдання

Реалізувати статистичний аналіз (z-тест, t-тест, bootstrap CI, Bayesian Probability to Win), SRM-діагностику, сегментований HTE-аналіз та інтерактивний вебінтерфейс на Streamlit

3 / 15

Постановка задачі

Функціональні вимоги до системи

01 Валідація CSV

Автоматична перевірка структури, типів та аномалій у вхідних даних

02 Статистичний аналіз

Z-тест, t-тест Велча, p-value, довірчі інтервали для бінарних та безперервних метрик

03 Bootstrap-оцінювання

Перцентильний CI для асиметричних метрик (revenue, AOV) без припущень про нормальність

04 Health-check / SRM

Автоматична діагностика Sample Ratio Mismatch та балансу user properties

05 Сегментований аналіз

HTE-аналіз за country, media source та іншими user properties

06 Структурований репорт

Primary / Secondary / Guardrail метрики з попередженнями у єдиному звіті

4 / 15

Основні результати

Математичний апарат системи

Z-тест для пропорцій
`z = (p̂B − p̂A) / √[p̂(1−p̂)(1/nA+1/nB)]`
Бінарні метрики: конверсія в підписку

T-тест Велча
`scipy.stats.ttest_ind (equal_var=False)`
Безперервні метрики: дохід, середній чек, довжина сесії

Довірчий інтервал
`CI = (p̂B−p̂A) ± z(α/2)·SE`
Інтервальна оцінка ефекту при будь-якому типі метрики

Beta-розподіл (байєсівський)
`Beta(conv+1, n−conv+1)`
"Ймовірність до перемоги" для бінарних метрик

Поправка Бонферроні
`α' = α / s`
Контроль кількості вибірки при сегментному аналізі

умова коректності розподілу

`srm_ok = max(|pA−0.5|, |pB−0.5|) ≤ δ-помилка розподілу груп`

δпомилка розподілу груп = 0.05

надійна оцінка ефекту (перцентильний)

`CI = [ θ̂*(α/2), θ̂*(1−α/2) ]`

B = 2 000–50 000 ітерацій

5 / 16

Основні результати

Потік даних у системі

Технологічний стек

Python 3.11 · Streamlit · Pandas · NumPy
SciPy · Statsmodels · Altair

Ключова властивість система підтримки рішень

Система формує структурований репорт — рішення про впровадження залишається за аналітиком

6 / 16

Основні результати

Чому рішення залишається за людиною

H_0: нова функція не змінює поведінку → H_1: зміна є → Система перевіряє статистично → Аналітик вирішує

✅ Запускаємо

H_1 підтверджена, перевірку якості пройдено, захисні показники не погіршились

❌ Не запускаємо

Негативний ефект або порушення якості тесту

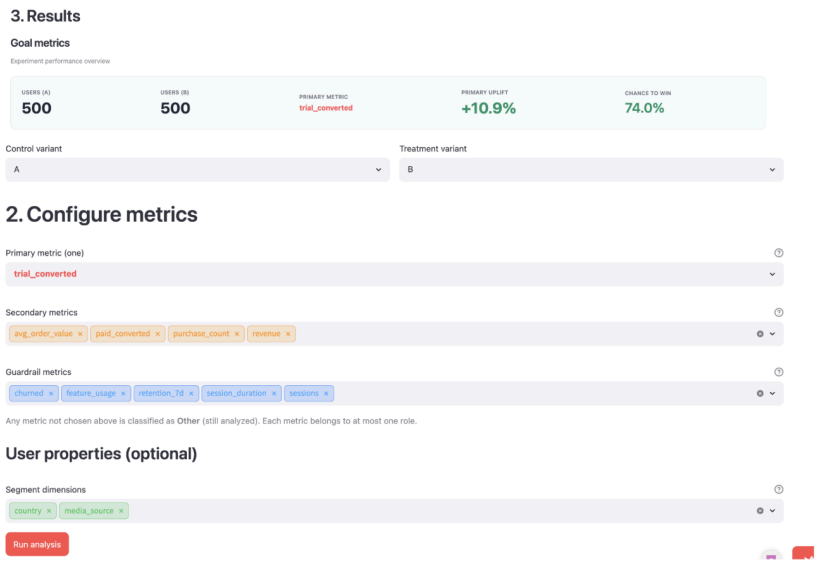
⚠️ Запускаємо з обмеженням

Загальний ефект позитивний, але в окремому сегменті ефект негативний

Статистика відповідає на питання «чи є ефект». Аналітик відповідає на питання «чи впроваджувати зміну у продукт».

Основні результати

Демо: конфігурація метрик та результати



Цільова метрика
конверсія у підписку — головний критерій рішення

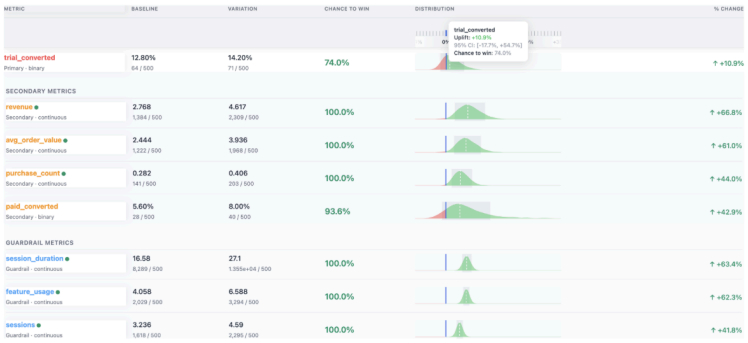
Вторинні метрики
дохід, середній чек, purchase_count

Захисні показники
churned, retention_7d, sessions

Сегменти
країна, джерело залучення

Основні результати

Структурований репорт та надійна оцінка ефекту



Репорт: ЦІЛЬОВА МЕТРИКА / ВТОРИННІ МЕТРИКИ / ЗАХИСНІ

Контрольна група · Тестова група · ймовірність перемоги
Розподіл оцінок · % зміни

надійна оцінка ефекту — чому важливо?

Дохід / Середній чек — правостороння асиметрія розподілу
Параметричний CI: ненадійний
Надійна оцінка ефекту: враховує реальну форму

95% CI конверсія у підписку: [–17.7%, +54.7%]

9 / 16

Основні результати

Перевірка якості експерименту

✓ Розподіл між групами коректний

A: 500 (50%) / B: 500 (50%)
dev_pp < 0.05 → розподіл коректний

✓ Характеристики груп збалансовані

Жодного перекосу
(threshold |Δ| ≤ 10%)

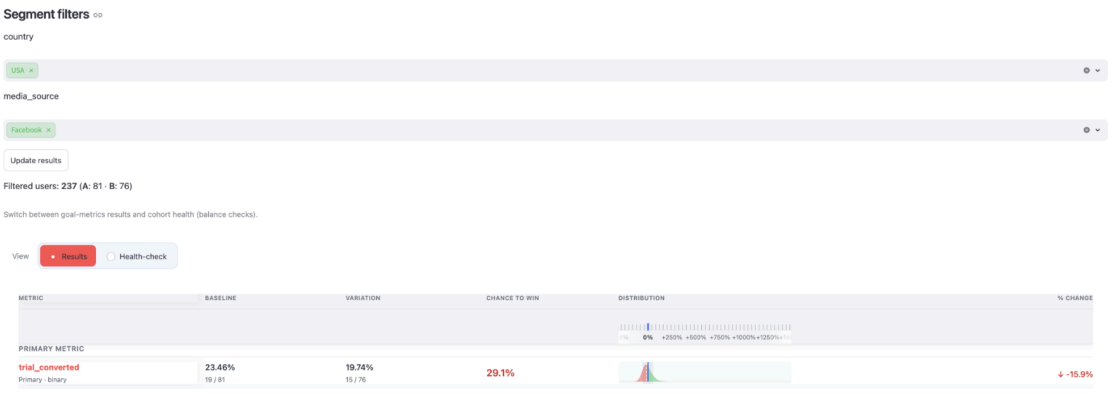
Bar-chart по media_source

Коректний розподіл трафіку між групами Facebook, Google, TikTok...

10 / 16

Основні результати

Сегментний аналіз — різний ефект у різних сегментах



Фільтр: USA · Facebook → 237 користувачів (A: 81, B: 76)

конверсія у сегменті: -15.9% при ймовірність перемоги 29.1% — ефект значно нижчий у сегменті, ніж у загальній вибірці (+10.9%)

Наукова новизна

Наукова новизна результатів

1 **Комплексна система підтримки рішень для A/B аналізу**

Вперше реалізовано єдину систему, що поєднує надійна оцінка ефекту, перевірку коректності розподілу груп та Захисні показники-метрики без вимог до технічної інтеграції — на відміну від усіх наявних комерційних рішень

2 **Формалізація структури репорту**

Формалізовано вектор аналітичного репорту $\phi(m) = \{n\_control, n\_treatment, value\_control, value\_treatment, приріст, статистичний критерій, probability\}$ та умова коректності розподілу з пороговим значенням $\delta = 0.05$

3 **Байєсівська оцінка для бінарних метрик**

Реалізовано Ймовірність перемоги через Beta-розподіл Beta(conv+1, n−conv+1) як альтернативу стандартному статистичному критерію для інтуїтивної інтерпретації результатів

4 **Аналіз різниці ефекту між сегментами без написання коду**

Запропоновано підхід до автоматичного сегментного аналізу з поправкою Бонферроні $\alpha' = \alpha/s$ через інтерактивний інтерфейс без необхідності написання аналітичного коду

Висновки

Висновки

1. Розроблено вебзастосунок системи підтримки рішень на Streamlit з повним статистичним апаратом: z-тест, t-тест Велча, надійна оцінка ефекту (2 000–50 000 ітерацій), байєсівська ймовірність перемоги через Beta-розподіл

2. Реалізовано модуль перевірки якості: перевірка коректності розподілу груп (поріг δ = 5%) та аналіз балансу характеристики користувачів між групами

3. Реалізовано аналіз різниці ефекту між сегментами за country, media source та характеристики користувачів без написання аналітичного коду

4. Система перевершує комерційні аналоги (Optimizely, VWO) за функціональністю при нульовому порозі входу — достатньо завантажити файл з даними

Економічна ефективність

Кількісна оцінка вигоди від автоматизації

⏱ Часова економія

45 хв економії на один тест → 90 люд-год/рік (≈11 робочих днів) при 10 тестах/міс

📊 Порівняння з аналогами

GrowthBook Pro: $480/рік (~20 160 грн)
Statsig Pro: $1 800/рік (~75 600 грн)
Amplitude Plus: $588/рік (~24 696 грн)

💰 Грошовий еквівалент

Junior (400 грн/год): 36 000 грн/рік
Middle (625 грн/год): 56 250 грн/рік
Formula: Et × Rate = 90 × Rate

🎯 Сукупний ефект

Консервативно: 56 160 грн/рік
Реалістично: 131 850 грн/рік
(час + заощаджена підписка)

Конкурентний аналіз

Функціональна ніша розробленої системи

✅ **Файл з даними — без технічної інтеграції**

Єдина серед аналогів система, що працює з довільним файл з даними без технічної інтеграції в продукт. Старт — миттєво.

✅ **Обов'язковий перевірка якості**

Перевірка якості експерименту обов'язкова на кожному кроці. У конкурентів — опціонально або лише на дорогих планах.

✅ **надійна оцінка ефекту + байєсівський метод**

Надійна статистична оцінка для показників з нерівномірним розподілом (дохід, середній чек). Конкуренти на базових планах — лише спрощені методи.

⚠️ **Обмеження системи**

Відсутній механізм рандомізації трафіку, збереження стану між сесіями та пакетна обробка — свідомі рішення для архітектури системи підтримки рішень.

15 / 16

Дякую за увагу!

Черток М.О. · КА-22 · 2026

Науковий керівник: Левенчук Людмила Борисівна

Система підтримки прийняття рішень для аналізу A/B тестів

надійна оцінка ефекту · перевірка коректності розподілу груп · Захисні показники-метрики · аналіз різниці ефекту між сегментами · 1 файл — без технічної інтеграції

16 / 16