

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту)
Кафедра обчислювальної техніки
(повна назва кафедри)

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«Методи аналізу даних в комп'ютерних системах»

Виконав: студент 2-го курсу, групи КІ-20м
спеціальності 123 – Комп'ютерна
Інженерія

(шифр і назва напрямку підготовки, спеціальності)

Проценко В.О.

(прізвище та ініціали)

Керівник: к.т.н., доц каф ОТ

Черняк О. І.

(прізвище та ініціали)

« _____ » _____ 2022 р.

Опонент: к.т.н., доц каф ЗІ

Дудатьєв А.В.

(прізвище та ініціали)

« _____ » _____ 2022 р.

Допущено до захисту
д.т.н., проф. Азаров О.Д.

" " 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет

Факультет Інформаційних технологій та комп'ютерної інженерії

Кафедра Обчислювальної техніки

Рівень вищої освіти II-ий (магістерський)

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

обчислювальної техніки

д.т.н., проф. Азаров О. Д.

«____» _____ 2022 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра наук
зі спеціальності: 123 – «Комп'ютерна Інженерія»

(шифр – назва спеціальності)

08-23.МКР.005.00.000 ПЗ

Магістранта групи КІ-20м

(назва групи)

Проценко Василь Олегович

(прізвище, ім'я і по батькові)

1 Тема магістерської кваліфікаційної роботи: «Методи аналізу даних в комп'ютерних системах» керівник роботи Черняк О.І., к.т.н., доцент, затверджено наказом вищого навчального закладу від **8 лютого 2022 року № 11**

2 Строк подання роботи **02.06.2022.**

3 Вхідні дані: достовірність та точність результатів інтервальних методів апроксимації функції одної і двох змінних та інтерполяції за формулою Лагранжа.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) : вступ, сучасні методи аналізу даних в комп'ютерних системах, методи

збору даних з веб-середовища для розробки системи веб-аналітики, опис програмного та технічного забезпечення, дослідження ефективності, економічна частина.

5 Графічна частина: математичні моделі, вхідні та вихідні дані поліноміальної апроксимації залежностей інтервальних даних, вхідні та вихідні дані, графіки результатів тестування інтервального методу інтерполяції за формулою Лагранжа .

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Тарновський М.Г., доцент	21.06.2021	21.06.2021
5	Руда Л.П, доцент	21.06.2021	21.06.2021

7 Дата видачі завдання **21.06.2021**

8 Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів МКР	Строк виконання	Підпис
1	Постановка задачі	17.01.2022	
2	Огляд існуючих методів	31.01.2022	
3	Збір даних з веб-середовища	04.02.2022	
4	Вибір засобів розробки	14.02.2022	
5	Розробка програмного забезпечення	31.03.2022	
6	Вирішення задачі по обробці даних	04.04.2022	
7	Проведення А/В тестування	20.04.2022	
8	Розрахунок економічної частини	30.04.2022	
9	Оформлення пояснювальної записки та ілюстративного матеріалу	05.05.2022	
10	Виконання магістерської кваліфікаційної роботи	06.05.2022	
11	Перевірка якості виконання магістерської кваліфікаційної роботи та усунення недоліків	09.05.2022	

Продовження таблиці 2

12	Підписи супроводжувальних документів у керівника, опонента, нормоконтролера	06.06.2022	
13	Перевірка «антиплагіат»	06.06.2022	
14	Попередній захист	09.05.2022	

Студент _____

Проценко В.О.

Керівник _____

к.т.н., доцент Черняк О.І.

АНОТАЦІЯ

Проценко В.О. Веб додаток з інтерактивним графіком, що відображає середнє значення перебування користувачів на сайті. Магістерська кваліфікаційна робота зі спеціальності 123 – Комп'ютерна Інженерія, Вінниця: ВНТУ, 2022.

Запропоновано метод аналізу даних в комп'ютерних системах, яке дозволило визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки про пріоритетність у соціальному житті населення

У магістерській кваліфікаційній роботі розроблено веб додаток на react js, у середовищі розробки Web Storm, що відображає середній час перебування користувачів на сайті у визначенні проміжки часу, на основі даних про час та дату входу на сайт, а також час та дату виходу з сайту, усіх відвідувачів за певний період часу.

Для того, щоб вирахувати як можна точніші данні, і відобразити їх на графіку, у роботі веб-додатку використано метод інтерполяції Лагранжа, який буде розглянутий більш детально у теоретичній частині роботи.

Магістерська робота містить вступ, 5 розділів, висновки, перелік джерел посилань і додатки. Загальний обсяг роботи – 102 сторінок, з яких основний зміст викладено на 70 сторінках друкованого тексту, містить 15 рисунків, 23 таблиці. Перелік джерел посилань містить 21 найменування.

Ключові слова: веб-аналітика, методи аналізу даних в комп'ютерних системах, перебування користувачів на сайті, метод інтерполяції за формулою Лагранжа

ABSTRACT

Protsenko V.O. Web application with interactive graph, that shows average duration of time that user stays on some website. Bachelor's thesis in specialty 123 - Computer Engineering. Vinnitsa: VNTU, 2022.

A method of data analysis in computer systems has been proposed, which allowed to identify the main trends in the interests of users, analyze them at selected intervals and determine the effectiveness of social life

In this master thesis I've created web application with interactive graph, that shows average duration of time that user stays on some website. Application based on react js and was developed in WebStorm environment. Application using data of all website users, that were logged to the web site, their time and date of login and time and date of logout, during some period of time.

To show most correct data on graph, we are using Lagrange's Interpolation method, that will be described more specific in theoretical part of master thesis.

The master's thesis contains an introduction, 5 chapters, conclusions, a list of sources of references and appendices. The total volume of the work is 102 pages, of which the main content is presented on 70 pages of printed text, contains 15 figures, 23 tables. The list of reference sources contains 21 names.

Keywords: web analytics, methods of data analysis in computer systems, stay of users on the site, interpolation method according to Lagrange's formula.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

HTML – HyperText Markup Language

ПЗ – Програмне забезпечення

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

HTTP – HyperText Transfer Protocol

API – Application programming interface

Web – система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету

CSS – Cascading Style Sheets (Каскадні таблиці стилів)

XML – eXtensible Markup Language (розширювана мова розмітки)

БД – База даних

SPA – (Single Page Application) Односторінковий додаток

KPI – Ключові показники ефективності

ЗМІСТ

ВСТУП	5
1 СУЧАСНІ МЕТОДИ АНАЛІЗУ ДАНИХ В КОМП'ЮТЕРНИХ СИСТЕМАХ	8
1.1 Методи аналізу та обробки даних.....	9
1.2 Огляд існуючих методів апроксимації даних	15
1.3 Інтерполяційний метод Лагранжа.....	20
2 МЕТОД ЗБОРУ ДАНИХ З ВЕБ-СЕРЕДОВИЩА НА ОСНОВІ ІНТЕРПОЛЯЦІЇ ЛАГРАНЖА	24
2.1 Змістовна наповнення задачі	24
2.2 Структура візуалізації алгоритму	25
2.3 Обґрунтування використаної математичної моделі.....	30
3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	31
3.1 Засоби розробки.....	31
3.2 Архітектура програмного забезпечення.....	40
3.3 Візуалізація роботи програмного рішення.....	42
3.4 Вирішення задачі по обробці і графічному відображенню даних.....	44
4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ	51
4.1 Порівняльний аналіз з Google Analytics	51
4.2 Визначення ключових переваг над конкурентами.....	52
4.3 Проведення А/В тестування	53
5 ЕКОНОМІЧНА ЧАСТИНА	57
5.1 Опис ідеї проекту.....	57
5.2 Технологічний аудит ідеї проекту	59

					08-23.МКР.006.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>	Проценко В.О.				МЕТОДИ АНАЛІЗУ ДАНИХ В КОМП'ЮТЕРНИХ СИСТЕМАХ ПОЯСНЮВАЛЬНА ЗАПИСКА	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>	Черняк О.І.					3	89	
<i>Рецензент</i>	Дудатьєв А.В.					ВНТУ, гр. 2КІ-20м		
<i>Н.контр.</i>	Швець С. І.							
<i>Затвердж.</i>	Азаров О.Д							

5.3 Аналіз ринкових можливостей запуску проекту	60
5.4 Розрахунок витрат на здійснення науково-технічної розробки	72
5.5 Оцінка результатів науково-дослідної роботи	76
ВИСНОВКИ	78
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТОК А Технічне завдання.....	82
ДОДАТОК Б Лістинг програми (Головний клас програми)	86
ДОДАТОК В Лістинг програми (Клас з логікою роботи графіка)	87
ДОДАТОК Г Лістинг програми (Файл стилів програми).....	92
ДОДАТОК Д Лістинг програми (Файл конфігурації програми).....	94
ДОДАТОК Е Лістинг програми (Файл з даними про користувачів)	96
ДОДАТОК Ж Лістинг програми (Головний html файл програми).....	100
ДОДАТОК К Протокол перевірки навчальної (кваліфікаційної) роботи.....	102

					08-23.МКР.006.00.000 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дат		

ВСТУП

Актуальність теми дослідження полягає в тому, що у зв'язку з розвитком різних напрямів науки і техніки, виникла необхідність обчислення не тільки наближених рішень різноманітних задач, але й гарантованих оцінок їх близькості до точних рішень. Використання методів інтервальної математики дозволяє не лише представити опис невизначених даних, а й врахувати неточність результатів обчислювальних операцій, що значно полегшує подальший аналіз достовірності даних. Одним з прикладних аспектів застосування математичних моделей є інформаційні системи, які збирають та обробляють великі масиви даних, та використовуються для підтримки вибору певних управлінських рішень.

Сучасні методи аналізу масивів даних знаходять застосування в різних областях теоретичних і прикладних досліджень, у тому числі в Веб-аналітиці.

Сьогодні Інтернет є обов'язковою частиною соціального життя. Кількість інтернет-користувачів у світі перевищила 4,7 млрд. Здебільшого такому росту сприяла поява доступних смартфонів. Свої акаунти і доступ до Інтернету має більш як 60% населення Землі. За останній рік 1 мільйон людей щоденно вперше в житті зареєструвалися у соціальних мережах. Це означає, що кожен секунду додавалось 12 нових користувачів. Одночасно з тим й ринок електронної комерції виріс за 2021 рік на 27,6%. Однак, яке значення цієї інформації, отриманої від відвідувачів веб-сайту, і що можна дізнатись про її вивченні?

Відповіді на ці запитання дає веб-аналітика - це пряме відображення поведінки відвідувача на Веб-сайті. Отримуючи різні показники веб-аналітики, такі як кількість відвідувань та відвідувачів (окремих осіб), а також тривалість відвідування, можна створити основні показники ефективності (KPI) — що являє собою універсальну аналітичну модель, яка вимірює кілька показників один проти одного, щоб визначати основні тенденції у запитах відвідувачів. KPI використовують, щоб отримати поглиблене уявлення про поведінку відвідувачів на сайті.

Веб-аналітика є прямим відображенням поведінки відвідувачів на веб-сайті. Ця

інформація дає змогу підприємствам коригувати свої веб-сайти та бізнес-цілі, визначаючи сфери для вдосконалення, знаходячи нові інструменти для просування веб-сайтів в Інтернеті, тестуючи нові функції та, зрештою, збільшуючи власні прибутки.

Часто при аналізі даних доводиться отримувати рішення математичних задач у числовому вигляді. Для багатьох проблем відоме лише існування рішення, а не остаточна формула. Звідси впливає завдання перетворення однієї форми функціональної залежності в іншу - перетворення табличних даних (вимірів) в аналізі (побудова математичних моделей). Часто алгоритми наближених розв'язків ґрунтуються на тому, що вихідна математична задача замінюється деякими простішими або частіше серією більш простих задач. Рішення цих більш простих задач інтерпретуються як наближені рішення вихідної задачі. Тобто фактично використовується якась модель початкової задачі.

Даний проект охоплює вивчення найпоширеніших показників, різні методи їх збору, аналіз ключових показників ефективності, а також створення, у вигляді програмного рішення, що засноване на математичній моделі інтервального методу інтерполяції за формулою Лагранжа, сучасного інструмента для веб-аналітики.

Об'єктом дослідження є веб-аналітика.

Предметом дослідження є система аналізу поведінки людини в мережевому середовищі на основі методів і алгоритмів збору даних.

Метою роботи є надання нових можливостей з аналізу даних при проведенні веб-аналітики.

Для досягнення цієї мети необхідно вирішити такі задачі:

- проаналізувати сучасні методи аналізу даних в комп'ютерних системах;
- розробити метод збору даних з веб-середовища на основі інтерполяції Лагранжа;
- на основі проведених досліджень розробити програмне забезпечення веб-аналітики для аналізу поведінки користувача в веб-середовищі.

Для досягнення поставленої в роботі мети використовуються такі **методи дослідження**:

— розробка методів й алгоритмів збору даних з веб-середовища для створення програмного інструменту веб-аналітики;

— математичне моделювання інтервального методу інтерполяції за формулою Лагранжа у програмному середовищі.

Наукова новизна отримав подальший розвиток метод збору даних з веб-середовища що полягає у інтерполяції отриманих даних методом ЛаГранжа.

Практичне значення роботи полягає в тому, що розроблене програмне рішення дозволяє визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки про пріоритетність у соціальному житті населення (в т.ч. по регіонах), а для комерційних представників - розробляти бізнес-стратегії для просування сайтів та охоплення нової аудиторії, базуючись безпосередньо на числових даних отриманих у ході дослідження

Особистий внесок здобувача:

Основні результати роботи отримані автором самостійно.

Апробація результатів роботи здійснена в доповіді на Всеукраїнська науково-практична інтернет-конференція студентів аспірантів та молодих науковців **МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ**

Матеріали роботи доповідались та опубліковувались:

Проценко В.О. Методи аналізу даних в комп'ютерних системах / В.О. Проценко, О. А. Рижій, М. Г. Тарновський // Тези доповіді. XLIX регіональної науково-технічної конференції професорсько-викладацького складу, співробітників та студентів ВНТУ з участю працівників науково-дослідних організацій. Вінниця 2022р.

Режим

доступу:

<https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/view/16216>

1 СУЧАСНІ МЕТОДИ АНАЛІЗУ ДАНИХ В КОМП'ЮТЕРНИХ СИСТЕМАХ

З розвитком сфери інформаційних технологій, інформаційні системи стали невід'ємною частиною технічної бази більшості підприємств. Застосування сучасних інформаційних систем підприємства дозволяє оптимізувати процеси його діяльності.

На даний момент ринок пропонує безліч товарів та послуг. Для того, щоб підприємство успішно вписувалося в рамки конкуренції, необхідно адаптувати всі його внутрішні процеси. Оскільки інформаційні системи є частиною підприємства та частиною цих процесів, то вони також можуть зазнавати змін.

Основна складова інформаційної системи – це інформація. З цієї причини існує необхідність у розумінні того, як працювати з нею, для отримання на виході необхідного результату. Оскільки інформаційна система — це програмний продукт, механізм сприйняття інформації матиме специфічний вигляд [1]. Необхідно вказати «машині», як і яку обробляти інформацію. Для вирішення цього завдання виділяють низку критеріїв до інформації, що надається:

- однорідність;
- обмеження обсягу даних;
- цінність даних;
- корисність.

Інформація має бути представлена в однорідному вигляді, тобто містити загальну структуру. Насамперед варто відзначити, що інформація може бути представлена в різних типах: наприклад, текстова, цифрова або змішана. Обумовлюючись конкретним завданням, необхідно, щоб система отримувала той чи інший конкретний тип даних.

Необхідно зрозуміти з яким обсягом даних доводиться працювати, оскільки, якщо інформації багато, то кількість інформаційного шуму може бути досить

великим. Великі показники інформаційного шуму зводять нанівець доцільність роботи з даними.

Щоб визначити цінність даних, необхідно зрозуміти, наскільки дані взаємопов'язані між собою. У випадку, коли інформація не містить взаємозв'язків, знижується ефективність роботи з такою інформацією і така діяльність може не призвести до потрібних результатів.

Інформація повинна мати корисність. Це означає, що, проаналізувавши інформацію, висновки призведуть до конкретного значення і застосовуватимуться практично.

Наприклад, отримуючи різні показники веб-аналітики, таких як кількість відвідувань та відвідувачів (окремих осіб), а також тривалість відвідування, можна створити універсальну аналітичну модель, яка надає можливість отримати поглиблене уявлення про поведінку відвідувачів на сайті. Ця інформація дозволяє комерційним компаніям коригувати свої веб-сайти та бізнес-цілі, виявляючи області для вдосконалення, знаходження нових інструментів онлайн-просування сайту, тестування нових функціональних можливостей та, зрештою, збільшення власних прибутків.

1.1 Методи аналізу та обробки даних

Дані, які можуть бути використані для аналізу, бувають чотирьох типів[2]:

- чисельні дані (вартість товару; 100 гривень);
- інтервальні дані (частка ринку підприємства; 5%);
- рангові дані (лояльність споживача; напій Coca-Cola подобається більше ніж Pepsi-Cola);
- номінальні дані (професія споживача; вчений, військовий, лікар).

Усі дані, які підходять під один із цих типів, можуть бути проаналізовані за допомогою формальних методів.

Будь-який набір даних може бути адекватно представлений комбінацією перелічених типів.

Основною метою веб-аналітики є збір та аналіз веб-трафіку та моделей використання веб-програм[3]. Найпоширенішим способом вивчення цих даних є використання розмірної моделі, яку було обрано й для даного проекту. Згідно з нею основні типи параметрів включають час, зміст, розташування, інформацію про користувача клієнта (наприклад, операційну систему, тип браузера). Дані з додатків зазвичай вбудовуються в HTTP-запити. Інформація надсилається на сервер для обробки за допомогою веб-маяків або веб-служб. Цей метод використовують й світові компанії такі як Google Analytics і Open Web Analytics.

На сторінці можна отримувати аналіз даних в реальному часі за допомогою налаштування певних критеріїв, таких як:

- тип аналізу (кількість, сума, мінімум, максимум, унікальні, процентна ставка і т.п.);
- тип події (click, pageview або будь-яке зазначене);
- timewrew — часові рамки;
- фільтр — налаштовуються конкретні параметри, що цікавлять користувача (= , > , < , ≠ , в списку, не в списку, в певному радіусі по геопозиції);
- інтервал — графік вибраних даних на проміжку часу.

Існують наступні сучасні та актуальні методи аналізу даних в інформаційних системах :

- кластерний аналіз;
- факторний аналіз;
- нейронні мережі;
- дерева рішень;
- регресійний аналіз;
- дискримінантний аналіз;

- кореляційний аналіз;
- апроксимаційний метод.

Термін «кластерний аналіз» (вперше ввів Трюон в 1939[4]) насправді включає в себе набір різних алгоритмів класифікації.

Загальне питання, що ставиться дослідниками у багатьох областях, полягає в тому, як розбити дані на групи з близькими значеннями параметрів.

Так, для веб-аналітики існують два основних типи даних: факти, дані вимірювань та вимірювальні дані, що описують факти з різних аспектів і рівнів. Дані фактів стосуються переважно кількості та часу використання. Найбільш основною мірою є перегляд сторінки, який є єдиним запитом для веб-сторінки.

У разі наявності великої кількості параметрів (більше 100) має сенс згрупувати параметри та аналізувати вже не кожен параметр окремо, а групи параметрів як єдиний комплексний параметр (фактор).

В основі факторного аналізу лежить ідея про те, що за складними взаємозв'язками явно заданих ознак стоїть відносно простіша структура, що відображає найбільш суттєві риси досліджуваного явища, а «зовнішні» ознаки є функціями прихованих загальних факторів, що визначають цю структуру.

Наприклад, при діагностуванні енергетичного обладнання штатними контрольно-вимірювальними приладами та обробці цих даних за допомогою, наприклад, програмних засобів можна прогнозувати відхилення роботи пристроїв задовго до їх зовнішнього прояву[4].

Інформацію про порушення, що виникають у процесі роботи енергетичних об'єктів, дають кореляційні зв'язки між різними експлуатаційними параметрами (температурою газу, температурою мастила в підшипнику турбіни та ін.) та загальними факторами (навантаженнями на обладнання).

При діагностуванні традиційними методами, тобто візуально за шкалою приладів, об'єкт виглядатиме працездатним, але кореляційні зв'язки можуть почати спотворюватися значно раніше, ніж з'являться відхилення від норми будь-якого

параметра. Такий аналіз особливо ефективний в історичній перспективі, на великих масивах зареєстрованих параметрів, коли спостереження фіксують стан пристрою протягом певного часу. Збільшення середньої відстані між факторними навантаженнями (при повній відповідності поточного стану пристрою до вимог технічної документації) завжди показує, що в роботі пристрою вже спостерігаються серйозні відхилення. При ручному веденні експлуатаційних журналів подібний детальний аналіз неможливий, а часто й самі записи не зберігаються. Внаслідок цього не вдається визначити передвідмовний стан обладнання, що призводить до серйозних аварій. Таким чином, виявлення внутрішніх закономірностей процесів, що відбуваються у складних системах різної фізичної природи (технічних, соціальних, біологічних, психологічних), дозволяє виявляти зміни значно раніше, ніж вони виявляються.

Метод нейронних мереж набув поширення у різних областях, де потрібно знайти значення невідомих змінних за відомими даними вимірювань (завдання регресії, класифікації, аналіз часових рядів). Модель нейронної мережі потребує досить великих масивів даних і особливо ефективна у випадках, коли закономірності між змінними дуже складні, наприклад:

- прогнозування фінансових часових рядів (зокрема цін на акції);
- оцінка кредитоспроможності за анкетними даними позичальника;
- управління виробничими процесами з безперервним регулюванням параметрів, що управляють;
- лінгвістичний аналіз (синтез та розпізнавання мови); обробка зображень та ін.

Дерево рішень – це метод представлення правил в ієрархічній послідовній логічній структурі, що дозволяє асоціювати об'єкт або ситуацію на вході з одним або кількома вихідними (термінальними) вузлами [5]. Під правилом розуміється логічна конструкція, подана у вигляді «якщо... то».

Метод дерев рішень може допомогти при прийнятті складного рішення, на яке впливають десятки параметрів.

Дерева рішень широко застосовуються в багатьох сферах діяльності:

- оцінка кредитоспроможності клієнтів банку під час видачі кредитів;
- випробування без руйнувань (наприклад, перевірка якості зварювання) контроль якості продукції (виявлення дефектів);
- діагностика різноманітних захворювань;
- аналіз будови амінокислот;
- компанія McKinsey використовує дерева рішень (issue tree, термін McKinsey) для консультацій своїх клієнтів.

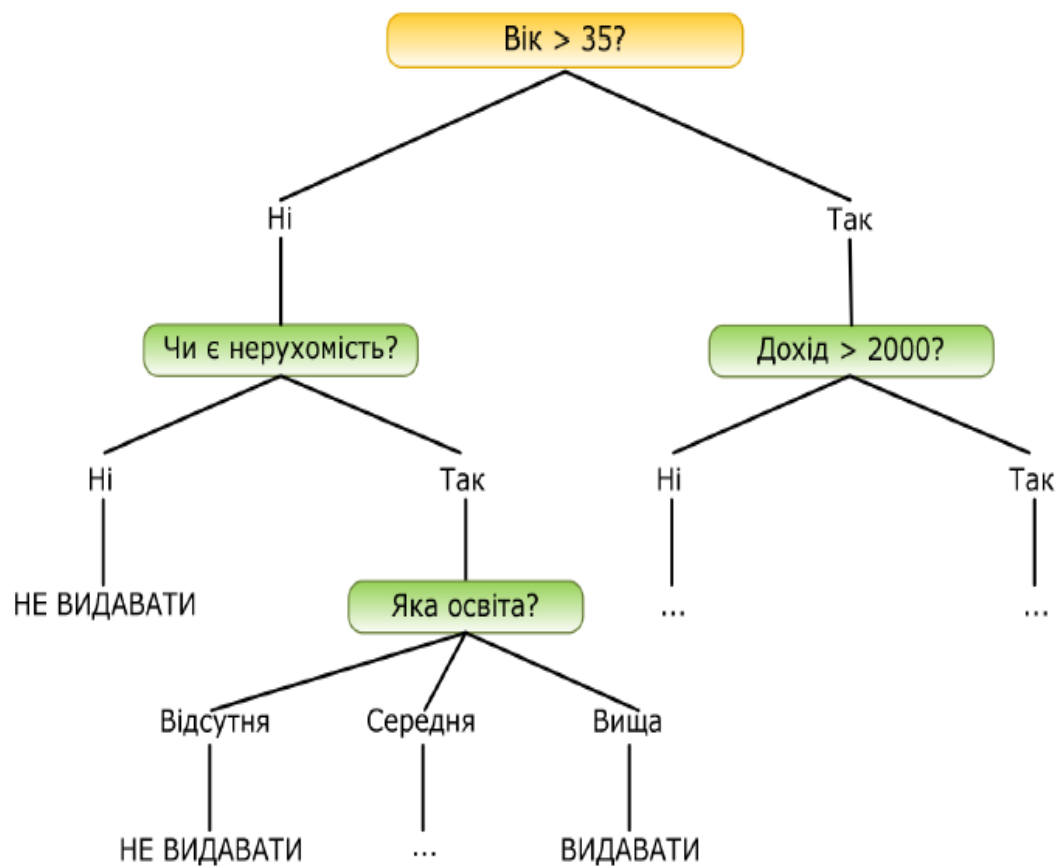


Рисунок 1.1 — Дерево рішень

Це далеко не повний список областей, де можна використовувати дерева рішень. Не досліджено ще багато потенційних сфер застосування цього інструменту.

Основною метою регресійного аналізу є визначення наявності і характеру зв'язку між змінними (у найпростішому випадку будується залежність $y(x)$, виходячи з приблизної форми кривої).

Регресійний аналіз широко використовується в офісному пакеті Excel, який надає можливість досліджувати як лінійні, а й інші, складніші залежності (в Excel це називається побудовою ліній трендів).

Дискримінантний аналіз – це статистичний інструмент, який використовується для визначення того, які змінні мають використовувати отриманий набір даних.

Кореляційний аналіз дозволяє судити про поведінкову схожість різних змінних [6]. У найзагальнішому вигляді прийняття гіпотези кореляції означає, що зміна значення змінної А буде співпадати з пропорційною зміною значення В: якщо обидві змінні збільшуються, кореляція позитивна; якщо одна змінна збільшується, а інша зменшується, то кореляція від'ємна.

Дивлячись на кореляції, вони намагаються визначити різницю між двома показниками в одній вибірці (наприклад, між зростом і вагою дитини або між IQ та академічними досягненнями) або між двома різними вибірками (наприклад, при порівнянні близнюків). взаємозв'язок між ними, і якщо так, то чи супроводжується збільшення одного показника збільшенням (позитивна кореляція) чи зменшенням (негативна кореляція) іншого.

Завдання аналізу даних попиту та пропозиції для вироблення оптимальних ринкових стратегій та скорочення витрат підприємства вирішується у різний спосіб. У тому числі використання спеціалізованого програмного забезпечення для прикладних та дослідницьких додатків. Згідно з традиційним підходом, прийнятим в економічній теорії, ринкові характеристики товару визначаються у двовимірній системі координат «ціна — кількість», не враховуючи безліч нецінових факторів.

У той самий час конкурентна боротьба змушує звертати особливу увагу чинник якості товару. При цьому поняття якості закладаються практично всі нецінові

характеристики товару: надійність, довговічність, технологічність, екологічність, соціальні та інші фактори.

1.2 Методи апроксимації даних

Завдання на апроксимацію сьогодні є актуальною темою майже для всіх технічних завдань. Від вибору типу апроксимації значною мірою залежать кількісні характеристики та якісний характер опису досліджуваного об'єкта.

У більшості випадків метою апроксимації є відновлення втрачених вихідних даних, але в технічній діагностиці важливо передбачити поведінку майбутніх вимірюваних параметрів, щоб оцінити залишкові ресурси. Однак у більшості технічних систем проблема прогнозування залишається невирішеною, оскільки існуючі математичні інструменти та 18 алгоритмів апроксимації в деяких випадках неефективні і не можуть знайти аналітичні описи вимірюваних величин з достатньою швидкістю та точністю.

Тому розробка нових методів апроксимації даних вимірювань є надзвичайно важливим і актуальним завданням у прогнозуванні багатокласових діагностичних систем.

Завдання на апроксимацію сьогодні є актуальною темою майже для всіх технічних завдань. Від вибору типу апроксимації значною мірою залежать кількісні характеристики та якісні властивості опису досліджуваного об'єкта.

Апроксимації дозволяють вивчати числові характеристики та якісні властивості об'єктів, зводячи завдання до вивчення більш простих або зручніших об'єктів (наприклад, тих, чиї характеристики легко обчислити або властивості яких відомі).

Часто характеристики багатьох складних процесів і явищ отримують експериментальним шляхом, і рідко вдається знайти їх теоретично. Щоб вивчити ці процеси, спочатку необхідно відобразити ознаки в математичній формі, придатній для обчислень [7].

Простий і досить точний спосіб – представити ознаки в таблиці. Цей метод полегшує використання електронно-обчислювальної машини (комп'ютера) для аналізу процесу, параметрів і функцій для створення двовимірного масиву в сховищі. У деяких випадках характеристики реальних процесів і явищ бувають комплексними і представлені графічно. Часто буває незручно безпосередньо застосовувати експериментальні дані у вигляді таблиць і графіків, і є перевага опису даних за допомогою простих аналітичних відносин, які принаймні якісно відображають характер розглянутих залежностей [8]. У цьому випадку необхідно вирішити задачу апроксимації, тобто замінити комплексну функцію (побудовану на основі експериментальних даних) на наближений аналітичний вираз. Тому, якщо дослідження проводити не чисельно, а аналітично, необхідно вибрати дуже просту апроксимаційну функцію, яка з достатньою точністю відобразатиме всі важливі особливості експериментально отриманих властивостей.

Приблизна загальна задача складається з двох окремих задач [9]:

- вибрати клас апроксимаційної функції, що найкраще описує вихідні дані;
- визначити значення постійного коефіцієнта, що міститься у функції апроксимації (визначити коефіцієнт апроксимації).

Для вибору класу апроксимуючої функції, необхідно дотримуватись наступних вимог [9]:

- простота функції (у сенсі математичних операцій і реалізації на ЕОМ);
- достатня точність (похибка апроксимації повинна бути одного порядку з розкидом параметрів характеристик окремих реалізацій в ансамблі реалізацій);
- наочність, яка дозволяє судити про зміну коефіцієнтів апроксимації при зміні характеристик процесу;
- ясність розуміння процесів в явищі і виявленні властивостей і характеристик, що представляють інтерес у конкретному випадку.

Тому функція, яка апроксимує властивість, базується або на фізичному зображенні досліджуваного процесу, або чисто формально, на основі зовнішньої

подібності властивості до графічного представлення функції [7]. Вимоги до функції апроксимації суперечливі: щоб забезпечити хорошу якість апроксимації, вона повинна бути відносно простою для подальшого використання. [10].

Застосування апроксимації функцій досить різноманітні: пошук математичних описів для дискретних даних, тобто встановлення зв'язків між вибірками вимірюваного сигналу для подальшого прогнозування можливих майбутніх значень вимірюваних значень; відновлення передачі сигналу через опору в проводах або даних, втрачених через до втрати пакетів під час передачі оптичної лінії зв'язку; відновлення вихідних даних під час кодування та декодування сигналу; розпізнавання та покращення зображення; відновлення під час архівування та стиснення даних. Крім того, дуже важливим завданням є розробка нових методів для апроксимації даних вимірювань при прогнозуванні багатокласових діагностичних систем.

Вибір апроксимації має ґрунтуватися на конкретному завданні дослідження [11]. Загалом, чим простіше рівняння, що використовується для апроксимації, тим краще опис кореляції. Тому важливо враховувати, наскільки значним є відхилення конкретної величини від тенденції і чим воно спричинене. При описі кореляції емпірично визначених значень більшу точність можна отримати, використовуючи більш складні багатопараметричні рівняння. Однак у певній серії емпіричних даних немає сенсу намагатися передати випадкові відхилення числових значень з максимальною точністю.

Важливіше визначити загальну закономірність, яка в даному випадку є найбільш логічною і з прийнятною точністю, представлена двопараметричним рівнянням статичної функції. Тому, обираючи метод апроксимації, дослідник завжди йде на компроміс: вирішує, чи доцільно і доцільно в даному випадку «пожертвувати» деталями, таким чином. Тому функція, яка апроксимує властивість, базується або на фізичному зображенні досліджуваного процесу, або чисто формально, на основі зовнішньої подібності властивості до графічного представлення функції [7]. Вимоги до функції апроксимації суперечливі: щоб забезпечити хорошу якість апроксимації,

вона повинна бути відносно простою для подальшого використання, наскільки узагальнено слід описати залежність зіставлюваних змінних. Замасковане випадковими відхиленнями емпіричних даних від загальних законів, ця апроксимація також дозволяє вирішувати багато інших важливих завдань:

- формалізація виявлених залежностей;
- знаходження невідомих значень залежних змінних шляхом інтерполяції

або, якщо можливо, екстраполяції.

Умовно апроксимації можна розділити на два типи (рис. 1.2) [12]:

- сувора теорія математичної апроксимації;
- фізична (технічна) апроксимація.

Суворі математичні теорії апроксимації включає наступні методи апроксимації [12]:

- поліноми (многочлени);
- сплайнам;
- відрізки ряду Фур'є;
- ортогогальний поліном;
- граничні задачі для самих функцій.

Математична теорія наближень була заснована як фундаментальна, глобальна теорія наближень, яка може не знадобитися для вирішення практичних завдань поточного застосування. Це може бути пов'язано з втратою актуальності проблеми з часом, або через складність теорії (приблизна функція), або через велику кількість орієнтовних коефіцієнтів.

Менш суворі апроксимації — фізичні (технічні) наближення або математичні моделі фізичних явищ, процесів (фізичні моделі), технічних пристроїв (його властивостей), сигналів (його параметрів), середовища, речовин тощо.

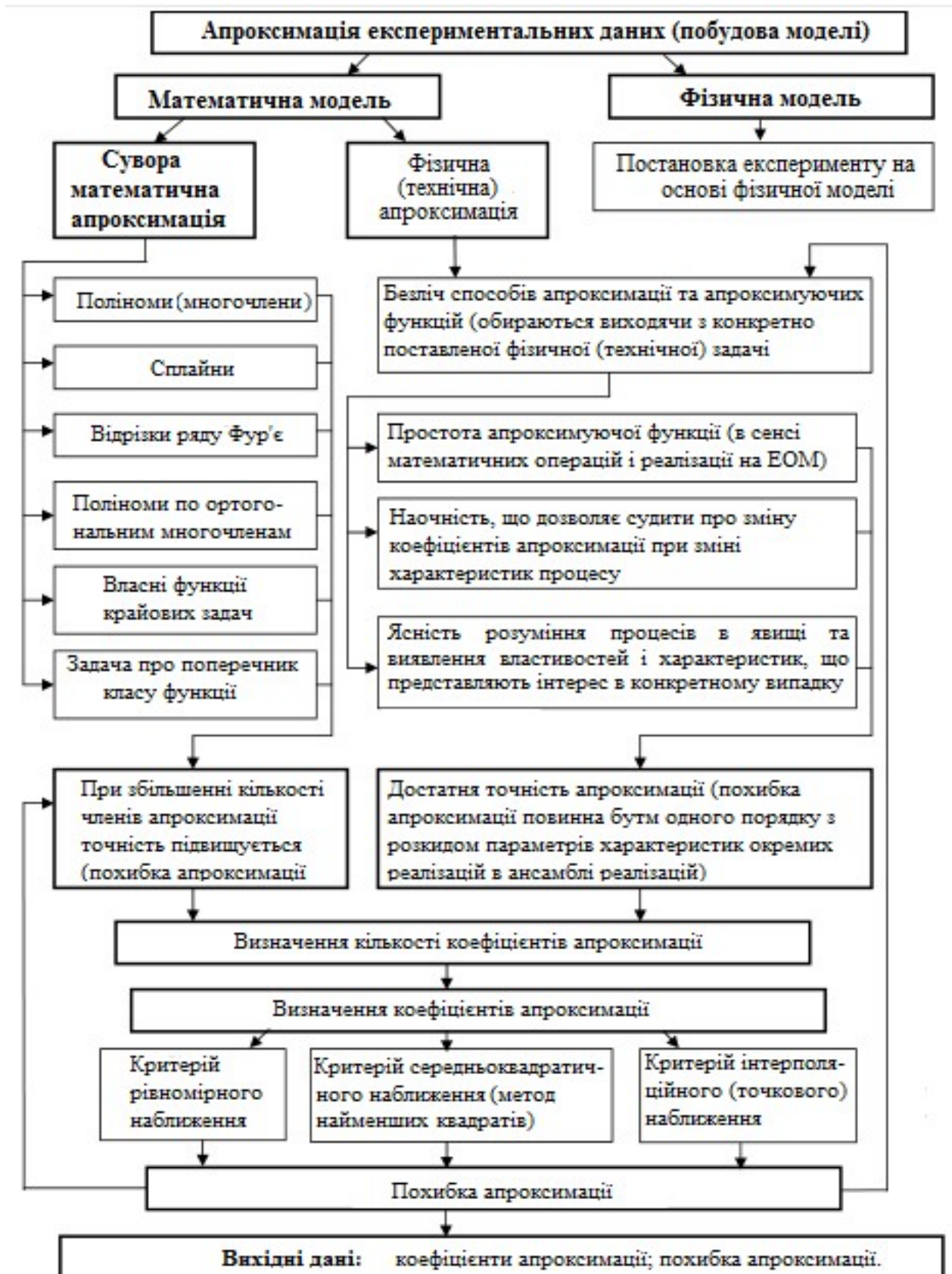


Рисунок 1.2 — Класифікація методів апроксимації експериментальних даних і побудова моделей

Фізична (технічна) апроксимація включає багато методів апроксимації та апроксимації функцій, які вибираються відповідно до конкретного фізичного (технічного) завдання.

За допомогою фізичної (технічної) апроксимації вирішується цілий спектр завдань, актуальних на даний момент часу, що пов'язані з рядом конкретних проблемам і питань прикладного (технічного) характеру.

Визначення коефіцієнтів апроксимації тісно пов'язане з необхідною точністю [13]. Для визначення точності потрібен критерій наближення, зазвичай використовують критерії рівномірного, середньоквадратичного і інтерполяційного (точкового наближення). Метод найменших квадратів можна використовувати коли число заданих точок перевищує число визначених коефіцієнтів апроксимації. При цьому середньоквадратична помилка буде мінімальною. Якщо необхідна висока точність апроксимації, використовують метод найменших квадратів.

Цей метод вимагає громіздких обчислень, але для аналітичного визначення коефіцієнтів моделі (апроксимації) має конструктивний підхід. Метод найменших квадратів забезпечує найменшу суму квадратів відхилень значень апроксимуючої функції від значень вихідної функції в довільному числі точок, не пов'язаному з числом невідомих коефіцієнтів.

Зазначимо, що чим точніше модель, яка описує дане явище, тим буде вище точність аналітичного уявлення досліджуваного явища. Найменша кількість коефіцієнтів моделі і її простота, одна з вимог, які пред'являють щодо вибору при однаковій точності моделі. Виконання даних вимог сприяють зменшенню систематичної помилки і часу обробки експериментальних даних.

Підвищення точності апроксимації та отримання більшої кількості відкликів можна досягти за рахунок попереднього використання алгоритмів інтерполяції.

1.3 Інтерполяційний метод Лагранжа

Існує багато методів інтерполяції. Деякі методи інтерполяції дуже гнучкі та підходять для різних даних, тоді як інші більш обмежені та вимагають, щоб дані

відповідали певним умовам. Кожен із цих методів має власний набір параметрів, які дозволяють налаштувати його для певного набору даних [14].

У цьому розділі буде розглянуто перший і один з найпростіших методів інтерполяції — метод Лагранжа.

Інтерполяційний поліном Лагранжа — многочлен мінімальної степені, що приймає необхідні значення в даному наборі точок. Для $n+1$ пар чисел $(x_0, y_0), (x_1, y_1) \dots (x_n, y_n)$, де всі x_j різні, існує єдиний многочлен $L(x)$ ступеня не більше n , для якого $L(x_i) = y_i$.

Інтерполяційний многочлен будується у вигляді лінійної комбінації многочленів степені n :

$$L(x) = y_0 l_0(x) + y_1 l_1(x) + \dots + y_n l_n(x) \quad (1.1)$$

Необхідно обнулити кожен поліном $l_i(x)$ у всіх вузлах інтерполяції, крім одного (i) , який має дорівнювати одиниці.

Цим умовам при $i=0$ відповідає многочлен виду:

$$l_0(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} \quad (1.2)$$

По аналогії з (2.2) отримаємо:

$$l_1(x) = \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)}$$

$$l_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)}$$

$$l_i(x) = \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

$$\ln(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})} \quad (1.3)$$

Підставимо в (2.1) вирази (2.2), (2.3), отримаємо:

$$L(x) = \sum_{i=0}^n y_i \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} \quad (1.4)$$

Ця формула визначає інтерполяційний многочлен Лагранжа.

гідно з наведеною формулою, недоліками методу судження [15] є:

— оскільки ступінь полінома Лагранжа визначається кількістю вузлів (мінус 1), будь-яка спроба підвищити точність інтерполяції за рахунок збільшення кількості вузлів потребує збільшення ступеня полінома;

— формула розрахунку досить складна, кожен компонент формули є поліномом ступеня n ;

— якщо ступінь полінома вище 5, на кривій з'являється «брижі», відомі як ефект Рунге-Меррі.

В інтерактивному режимі ситуацію можна покращити, вибравши розташування вузлів відповідно до конкретних функцій, але цей процес досить незручний.

Хоча рівняння (1.4) є громіздким, однією з переваг формули Лагранжа є те, що її можна записати безпосередньо з заданої таблиці значень функції. Для цього враховуйте такі правила: формула містить стільки доданків, скільки вузлів у таблиці; кожен доданок є добутком малого коефіцієнта, що відповідає значенню y_i ; чисельник коефіцієнта при y_i містить суму x і всі вузли, крім x_i Добуток різниць зі знаменником точно повторює чисельник при $x = x_i$. Блок-схема алгоритму інтерполяції поліномом Лагранжа представлена на рис.1.3 [16].

Тому використання принципу інтервального аналізу є ефективним способом включення точного рішення задачі в результат розрахунку з урахуванням помилок

розрахунку, помилок округлення тощо. Також це ефективний підхід до вирішення задач з вхідними даними заданими в умовах невизначеності.

Виконання програмного рішення, що засноване на математичній моделі інтервального методу інтерполяції за формулою Лагранжа дозволили на основі узагальнення відомих підходів і використання нових положень запропонувати нові алгоритми та методики для реалізації у інтервальній математиці методів поліноміальної апроксимації функції однієї і двох змінних.

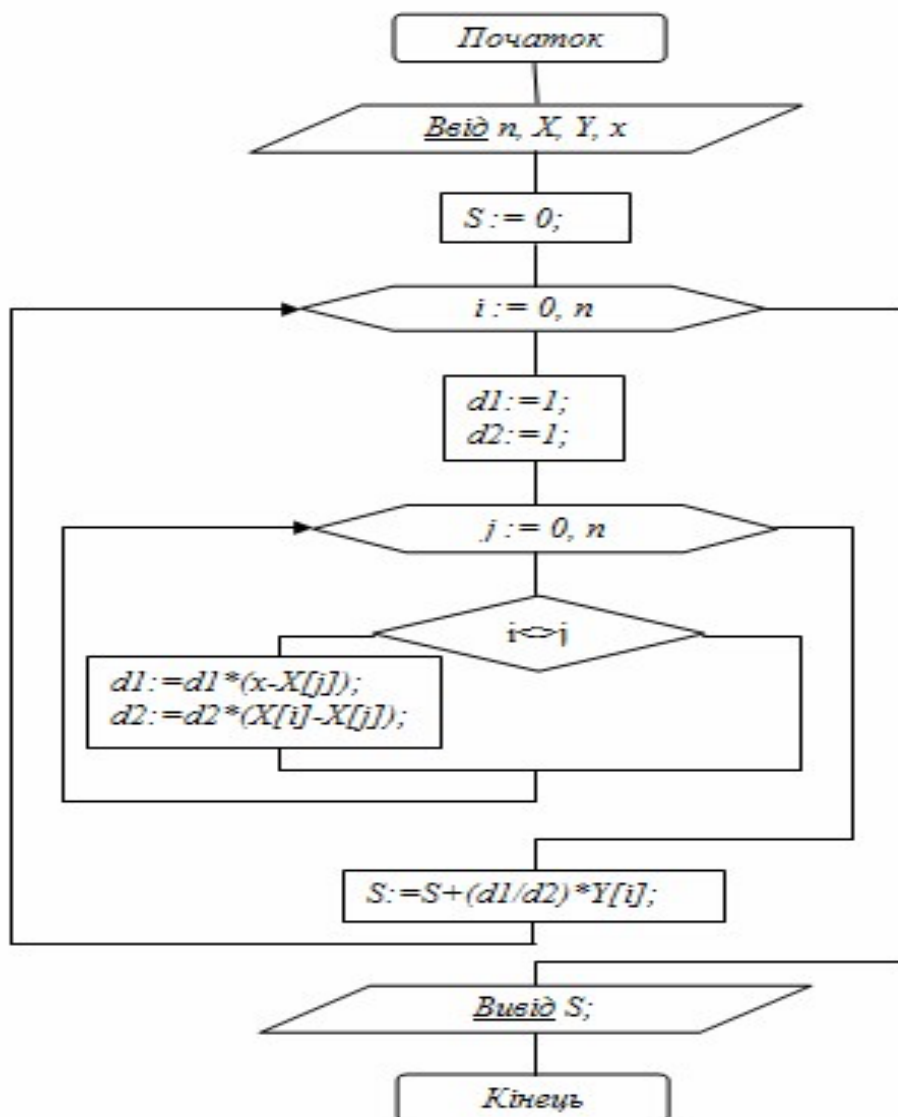


Рисунок 1.3 — Блок-схема алгоритму обчислення многочлена Лагранжа

2 МЕТОД ЗБОРУ ДАНИХ З ВЕБ-СЕРЕДОВИЩА НА ОСНОВІ ІНТЕРПОЛЯЦІЇ ЛАГРАНЖА

Розробка та реалізація системи аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з веб-середовища.

Створене програмне рішення дозволяє визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки про пріоритетність у соціальному житті населення (в т.ч. по регіонах), а для комерційних представників — розробляти бізнес-стратегії для просування сайтів та охоплення нової аудиторії, базуючись безпосередньо на числових даних отриманих у ході дослідження.

Для розробки методу необхідно розв'язати наступні задачі:

- визначити потребу в аналізі поведінки людини в мережі Інтернет;
- розробити імітаційну модель веб-середовища та поведінки користувача;
- виробити критерії оцінки результатів аналізу поведінки відвідувача веб-сайту;
- провести огляд існуючих технічних/програмних рішень (Лічильники, Google Analytics);
- сформулювати та обґрунтувати рішення щодо розробки системи аналізу поведінки;
- скласти план реалізації програмного продукту.

2.1 Змістовна наповнення задач

Фундаментальною основою веб-аналітики є збір та аналіз даних про використання веб-сайту. Сьогодні веб-аналітика використовується у багатьох галузях промисловості для різних цілей, включаючи моніторинг руху, оптимізацію електронної комерції, маркетинг / рекламу, веб-розробку, інформаційну архітектуру, поліпшення роботи веб-сайтів, веб-кампанії.

Послідовність та обґрунтування кроків для виконання задачі:

1) Спроекувати структуру бази даних на MySQL з усією інформацією, яку необхідно аналізувати.

2) Розробити серверний додаток в якості прокладки між клієнтською частиною та спроектованою базою даних.

3) Сегментувати трафік за активністю, залученістю, зацікавленістю і типу відвідувача.

4) Налаштувати візуалізацію даних з можливістю різної фільтрації за допомогою графіків, що пов'язаний з базою MySQL.

За зацікавленістю сегментуємо так:

- незацікавлені (сеанси тривалістю до 60 секунд);
- цікавляться (сеанси тривалістю 60-180 секунд);
- зацікавлені (сеанси тривалістю понад 180 секунд).

Всі звіти в даному файлі налаштовані на автоматичне оновлення раз на добу.

Розроблене програмне рішення має задовольняти наступні вимоги:

- візуалізувати обсяг трафіку і поведінкові показники по днях;
- виводити класифікацію відвідувачів у формі звіту;
- відображати сегментацію відвідувачів в формі діаграм;
- відображати зміни поведінкових показників;
- аналізувати цінність нового контенту;
- порівнювати показники окремих сторінок одного сайту.

2.2 Структура візуалізації алгоритму

Інформаційно-вимірювальні системи описуються значним числом показників якості. При цьому показники, як правило, суперечливі. Задача оптимального вибору технічних параметрів, сигналів і структур, при яких досягається достатньо ефективне значення необхідних показників ефективності при заданих обмеженнях, є завданням оптимального синтезу зазначених параметрів, сигналів та структур.

Щоб зрозуміти переваги аналізу веб-сайтів, потрібно спочатку зрозуміти показники — різні види доступної інформації про користувачів. Хоча показники можуть здаватися основними, коли вони зібрані, вони можуть бути використані для аналізу веб-трафіку та вдосконалення веб-сайту, щоб краще відповідати його трафіку. За даними Panalysis, яка є австралійською компанією Web Analytics, ці показники зазвичай відносяться до однієї з чотирьох категорій: використання сайту, рефералів (або, як відвідувачі прибули на ваш сайт), аналіз вмісту сайту, і забезпечення якості. У таблиці 2.1 наведено приклади типів показників, які можуть бути знайдені в цих категоріях.

Таблиця 2.1 — Класифікація показників

Site Usage	Referrers	Site Content Analysis	Quality Assurance
<ul style="list-style-type: none"> — Numbers of visitors and sessions — How many people repeatedly visit the site — Geographic information — Search Engine Activity 	<ul style="list-style-type: none"> — Which websites are sending visitors to your site — The search terms people used to find your site — How many people place bookmarks to the site 	<ul style="list-style-type: none"> — Top entry pages — Most popular pages — Top pages for single page view sessions — Top exit pages — Top paths through the site — Effectiveness of key content 	<ul style="list-style-type: none"> — Broken pages or server — Visitor response to errors

Хоча тип і загальна кількість показників залежать від різних постачальників аналітики, загальний набір базових показників, який є загальним для більшості користувачів, все ще існує. У таблиці 2.2 наведено вісім поширених типів інформації, які вимірюють, хто відвідує веб-сайт та що робить під час їх відвідування, пов'язавши кожен із цих показників з певними категоріями.

Даний показник складається з відношення кількості звернень, отриманих сторінкою, до підрахунку кількості осіб, які відвідують веб-сайт.

Таблиця 2.2 — Характеристика основних показників

Metric	Description	Category
Visitor Type	Who is accessing the Website	Site usage
Visit Length	The total amount of time a visitor spends on the Website	Site usage
Demographics and System Statistics	The physical location and information of the system used to access the Website	Site usage
Internal Search Information	Information on keywords and results pages viewed using a search engine embedded in the Website	Site usage
Visitor Path	The route a visitor uses to navigate through the Website	Site Content Analysis
Top Pages	The pages that receive the most traffic	Site Content Analysis
Referring URL and Keyword Analysis	Which sites have directed traffic to the Website and which keywords visitors are using to find the Website	Referrers
Errors	Any error that occurred while attempting to retrieve the page	Quality Assurance

Є два типи відвідувачів: ті, хто раніше відвідував сайт, і тих, хто цього не робив. Ця різниця визначається з точки зору повторних та нових відвідувачів. Щоб відстежувати відвідувачів таким чином, система повинна мати можливість визначати окремих користувачів, які мають доступ до веб-сайту; кожного окремого відвідувача називають унікальним відвідувачем.

В ідеалі, унікальним відвідувачем є лише один відвідувач, але це не завжди так (рисунок 2.1). Цілком можливо, що декілька користувачів отримують доступ до сайту з одного комп'ютера (можливо, на спільному комп'ютері або в публічній бібліотеці). Крім того, більшість аналітичних програм покладається на файли cookie для відстеження унікальних користувачів. Якщо користувач вимикає файли cookie у своєму браузері або очищає їх кеш-пам'ять, відвідувач буде вважатися новим щоразу, коли він або вона входить на сайт. Через це доцільним є відслідковувати унікальні відвідування або сеанси. Сеанс починається, коли користувач входить на сайт і закінчується, коли користувач виходить з сайту або після встановленої кількості часу бездіяльності (зазвичай 30 хвилин). Дані сеансу не залежать від файлів cookie та можуть бути легко виміряні. Оскільки виявлено меншу невизначеність за відвідуваннями, вона вважається більш конкретною та надійною, ніж унікальні відвідувачі.

Показник, який також іменується "Середня тривалість на сайті"), — це загальна кількість часу, який відвідувач витрачає на сайт протягом одного сеансу. Однією з можливих непорозумінь при використанні цього показника є обробка відсутніх даних. Це може бути викликано помилкою в зборі даних або сеансом, який містить лише один візит сторінки або взаємодію.

Мета вимірювання даних таким способом полягає в тому, щоб зберегти відсоток відвідувачів, які залишаються на Веб-сайті менш ніж за п'ять секунд, настільки низько, наскільки це можливо.

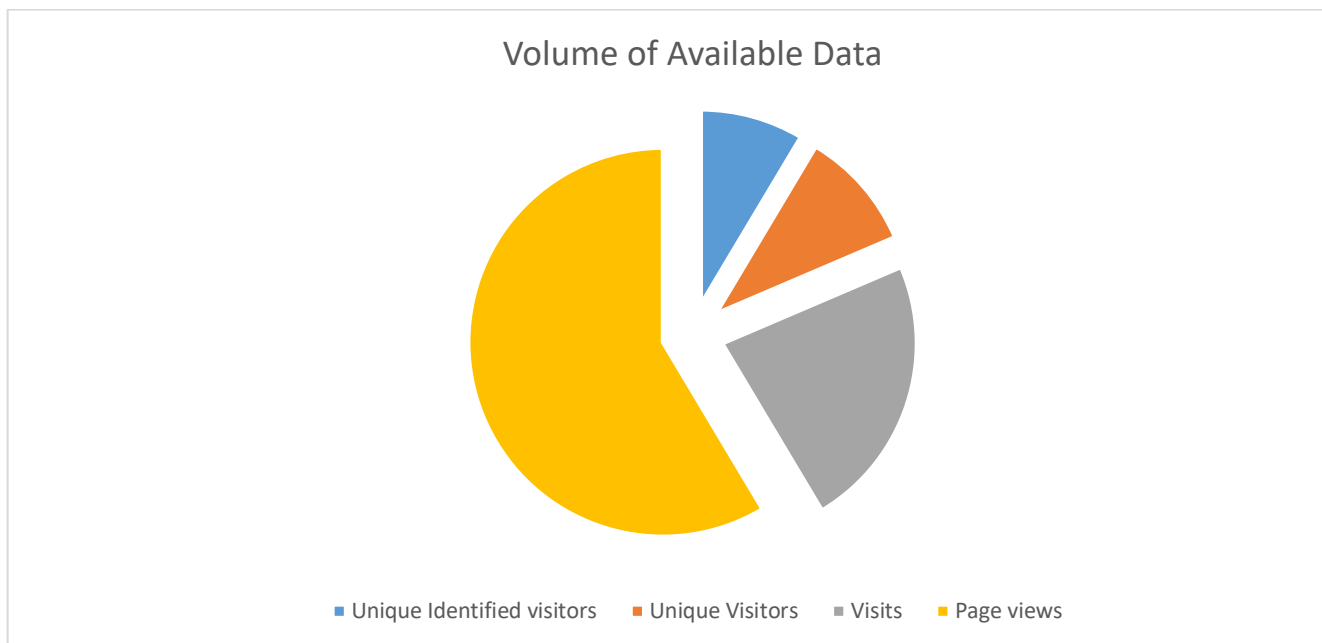


Рисунок 2.1 — Значення показників

Якщо відвідувачі залишаються на веб-сайті протягом такої короткої кількості часу, це зазвичай означає, що вони випадково прибули на сайт, або сайт не мав відповідної інформації. Поєднуючи цю інформацію з інформацією з реферерів та аналізу ключових слів, можна визначити, які сайти проводять цілеспрямований трафік, а також, які сайти слід віднести таких, що мають до низькоякісний трафік.

За умови, що веб-сайт містить у собі утиліт пошуку, то також можна виміряти обсяг внутрішньої інформації пошуку. Це включає в себе не тільки ключові слова, а й інформацію про те, які сторінки результатів відвідувачі визнали корисними.

Внутрішній пошук передбачає:

- визначення тенденцій споживачів;
- покращення персоналізованих повідомлень за допомогою власних слів клієнтів;
- визначення виникаючих питань обслуговування клієнтів;
- з'ясування, чи надано користувачам достатньо інформації для досягнення їх цілей;
- зробити персоналізовані пропозиції.

Шлях відвідувача — це шлях, який відвідувач використовує для навігації по веб-сайту. За винятком відвідувачів, які покидають сайт. Як тільки користувач заходить на сайт, кожен створює окремий шлях переглядів сторінок та дій під час перегляду сайту. Вивчаючи ці шляхи, можна виявити будь-які труднощі, пов'язані з переглядом конкретної ділянки сайту або виконанням певної дії (наприклад, здійснення транзакції або заповнення форми).

Однак, часто дії відвідувача є випадковими та нелогічними, і єдиним шляхом, який може надати точні дані про поведінку відвідувача, є шлях від однієї сторінки до сторінки, що безпосередньо слідує за нею. Іншими словами, єдиною сторінкою, яка впливає на поведінку відвідувачів на веб-сайті, є та, яку вони переглядають зараз. Цей метод аналізу стає все більш популярним, тому що компаніям легше аналізувати дані шляху в контексті, не маючи посилання на весь сайт, щоб вивчити поведінку відвідувачів.

В аналітиці існує три типи топ-сторінок: сторінка для входу, сторінка для виходу та найпопулярніші сторінки. Найпопулярніші сторінки вводу важливі, тому що перша сторінка переглядів відвідувача робить найбільше враження про веб-сайт. Знаючи верхню сторінку входу, можна переконатися, що сторінка має релевантну інформацію та надає належну навігацію важливим частинам сайту. Точно так само, виявлення популярних сторінок виходу полегшує виявлення збігів або відсутності вмісту.

Найпопулярніші сторінки — це ділянки веб-сайту, які отримують найбільший трафік. Цей показник дає уявлення про те, як відвідувачі використовують веб-сайт, а також про те, які сторінки надають найбільш корисну інформацію.

Помилки — наш останній показник. Помилки відстеження мають очевидну вигоду, оскільки вони можуть виявляти та виправляти будь-які помилки на Веб-сайті, але також корисно спостерігати за тим, як відвідувачі реагують на ці помилки. Чим менше відвідувачів, яких заплутали помилки на веб-сайті, тим менш імовірно, що відвідувачі виходять із сайту через помилку (рис.2.2).

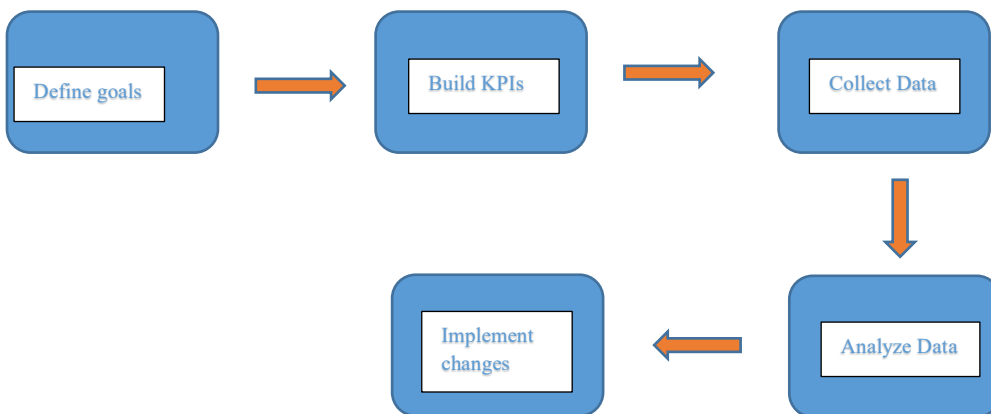


Рисунок 2.2 — Візуалізація процесу Веб-аналітики

2.3 Обґрунтування використаної математичної моделі

Математична модель використана для реалізації проекту ґрунтується на методиці Data Science, що спирається методи статистики. За її допомогою ми визначаємо чи виходить отримана величина за межі області допустимих значень даної метрики (так званий довірчий інтервал), чи являється зміна критичною.

В цілому напрямом Data Science дозволяє вирішити наступні завдання:

- керувати рішеннями, які знижують ймовірність несприятливого результату та мінімізують можливі втрати бізнесу;
- прогнозувати практично будь-які важливі показники. Можна використовувати для цього модель Бокса-Дженкинса (ARIMA).

3. ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмного забезпечення — це структура програми або комп'ютерної системи, яка включає компоненти програмного забезпечення, видимі властивості цих компонентів та зв'язки між ними [17]. Цей термін також застосовується до документування архітектури програмного забезпечення. Документування архітектури програмного забезпечення спрощує процес комунікації між зацікавленими сторонами, дозволяє фіксувати рішення, прийняті щодо проектування системи високого рівня на ранніх стадіях проектування, і дозволяє повторно використовувати дизайн і шаблони в інших проектах.

Вибір або створення архітектури залежить від конкретного завдання.

Наприклад, наскільки загальним є план програми, які модулі повинні існувати, яке ресурсне навантаження плану.

3.1 Засоби розробки

Під час створення програмного продукту було використано програмні засоби:

- середовище розробки JetBrains WebStorm;
- мова програмування Java Script;
- react JS для розробки веб-додатку;
- шаблон проектування архітектури додатку Model-View-Controller;
- текстовий формат JSON;
- chart.js для реалізації графіка.

Наш веб-додаток буде обробляти данні в json форматі, які він буде вчитувати з файла (В реальності ми б отримували ці данні з сервера, на який зберігаються данні про сесії користувачів). Після обробки даних ми будемо показувати графік, на якому ми будемо показувати середній час перебування користувачів, за певний період часу, з певним діапазоном (кожні 6 годин).

Так як ми будемо розробляти веб-додаток з графіком, то для розробки такого додатку найкраще підійде мова програмування Java Script, при цьому ми можемо

використати технологію React JS для більш комфортної та швидкої розробки веб додатку.

Для розробки веб-додатку будемо використовувати середовище програмування JetBrains WebStorm.

JetBrains WebStorm — інтегроване середовище розробки на JavaScript WebStorm забезпечує автодоповнення, аналіз коду на льоту, навігацію по коду, рефакторинг, зневадження та інтеграцію з системами управління версіями.

Важливою перевагою інтегрованого середовища розробки WebStorm є робота з проектами (у тому числі, рефакторинг коду JavaScript, що міститься в різних файлах і теках проекту, а також вкладеного в HTML) [18]. Підтримується множинна вкладеність (коли в документ на HTML вкладений скрипт на Javascript, в який вкладено інший код HTML, всередині якого вкладений Javascript) — в таких конструкціях підтримується коректний рефакторинг.

JavaScript — це мова програмування, що дозволяє зробити Web-сторінку інтерактивною, тобто такою що реагує на дії користувача.

Послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний Web-браузер. Іншими словами, код програми вбудовується в HTML — документ і виконується на боці клієнта. Для виконання програми не потрібно навіть перезавантажувати Web - сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі значення і, якщо значення не відповідають очікуванім, заборонити відправлення даних.

JavaScript — об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript.

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам [19].

Основні архітектурні риси:

- динамічна типізація;
- автоматичне керування пам'яттю;
- прототипне програмування;
- функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання непрограмістами. JavaScript має низку властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам:

- функції як об'єкти першого класу;
- об'єкти як списки;
- каррінг;
- анонімні функції;
- замикання, що додає мові додаткову гнучкість.

Незважаючи на схожий з Сі синтаксис, JavaScript у порівнянні з мовою Сі має корінні відмінності:

- об'єкти, з можливістю інтроспекції;
- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

У мові відсутні такі корисні речі, як:

- модульна система - JavaScript не надає можливості управляти залежностями та ізоляцією областей видимості;
- стандартна бібліотека — зокрема, відсутній інтерфейс програмування додатків по роботі з файловою системою, управління потоками вводу/виводу, базових типів для бінарних даних;

- стандартні інтерфейси до веб-серверів та баз даних;
- система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх.

Синтаксис мови JavaScript дуже нагадує синтаксис Сі і Java, семантично ж мова набагато ближче до Self, Smalltalk або навіть Ліспу.

В JavaScript :

- всі ідентифікатори Реєстрозалежні;
- в назвах змінних можна використовувати літери, підкреслення, символ долара, арабські цифри;
- назви змінних не можуть починатися з цифри;
- для оформлення однорядкових коментарів використовуються `/ /`, багаторядкові і внутрішньорядкові коментарі починаються з `/ *` і закінчуються `* /`.

ReactJs — це бібліотека JavaScript з відкритим кодом, яка використовується спеціально для побудови користувацьких інтерфейсів. Зазвичай використовується для односторінкових програм. Він використовується для обробки всіх переглядів програми для будь-яких веб-або мобільних додатків. ReactJS також використовується для повторного використання компонентів інтерфейсу. React дозволяє розробникам створювати веб-програми, які можуть змінювати ваші дані, не завантажуючи вашу сторінку. Основна перевага React JS полягає в тому, що він масштабований, простий та швидкий. Це також відповідає виду в шаблоні MVC. Зазвичай він виступає комбінацією бібліотек або фреймворків JavaScript [20].

Раніше створення динамічної веб-сторінки вимагало багато складного кодування. Для динамічних веб-сторінок знадобилися конкретні рядки HTML. Використання ReactJS — це вже не проблема. Він використовує JSX, який має певний синтаксис і дозволяє HTML-котирування та теги для надання певних підкомпонентів. На додаток до цього, він також підтримує побудову машиночитабельного коду та змішує компоненти в окремий файл змінної. У React регулярний JavaScript не використовується для створення його шаблонів.

Повторне використання компонентів є головною перевагою React JS. Навіть Facebook реалізував React, оскільки це полегшує повторне використання системних компонентів. Розробник може починати з звичайних компонентів, таких як прапорець, кнопка тощо. Після цього ви перейдете до компонентів обгортки, що складаються з внутрішньої логіки, що полегшує маніпулювання та визначення об'єктів, що використовуються. Це забезпечує узгодженість програми та полегшує підтримку коду. Багато компаній, таких як Walmart, як правило, використовують ReactJS. Оскільки у всіх компаній є такі компоненти, як елементи форми оплати, карусель із зображеннями, наприклад, сухарі. Тут враховується повторне використання компонентів. Для включення управління версіями та подальшої установки компонентів існують стандарти кодування, визначені для збереження рівномірності. Це також підвищує швидкість роботи програми, яка користується більшим утриманням клієнтів.

Програма Model-View-Controller (MVC) дозволяє розробляти, реалізовувати і тестувати кожен частину програми незалежно від будь-якої іншої, зберігаючи код організованим. Збереження організованого коду означає можливість швидко знайти те, що необхідно, щоб перевірити, швидко виправити, змінити і додати нові функції. Це також означає більш ефективний код і кращий спосіб повторного використання його для більш швидких додатків. Без вагомих підстав використовувати нову структуру, технологію або тренд багатьом розробникам важко, і насамперед тому, що вони не можуть знайти мотивацію для вивчення нової теми. Але тільки не у відношенні MVC, архітектура якої дуже важлива, а застосовувати методи MVC для веб необхідно.

Які ж основні типи функціональності архітектури?

Одною з найбільших переваг є те, що розробники розуміють і використовують структуру MVC для створення веб-додатків. З-за цієї узгодженості управління проектом між кількома розробниками стає простіше. В основному веб-додаток або частина програмного забезпечення слідує за структурою MVC. Якщо структура

представлена трьома основними типами функціональності, тоді зрозуміло, що це - MVC: Код моделі зазвичай відображає реальні речі. Цей код може містити необроблені дані або визначати основні компоненти програми. Наприклад, якщо користувач створював додаток Todo, код моделі визначав би, що таке «завдання» і, що таке «список», оскільки це основні компоненти даного додатка.

Вид, або подання — перегляд коду складається з усіх функцій, які безпосередньо взаємодіють з користувачем. Це код, який робить додаток красивим і в іншому випадку визначає, як користувач бачить і взаємодіє з ним. Контролер діє як зв'язок між моделлю та поданням, приймаючи користувальницький ввід і вирішуючи, що з ним робити. Це мозок програми і зв'язує модель і подання. Контролер вважають середнім рівнем». Він взаємодіє з користувачем, збираючи дані, контактує з моделлю, отримуючи необхідні дані, а потім з поданням, щоб відповісти користувачеві. Коли користувач виконує яку-небудь дію, він спочатку переходить до контролера. Він буде приймати будь-які дані, наприклад, `$_GET`, `$_POST` змінні в PHP, і визначати, що робити з цими даними. Коротше кажучи, моделі відносяться до обробки даних і розширеної функціональності. Тому завдання контролера в цій точці полягає в тому, щоб визначити, яку модель слід викликати, а потім відкрити відповідну функцію всередині цієї моделі. Після виклику функції він буде знаходити результат, зазвичай у змінному середовищі.

Модель — це просте уявлення про те, що користувач виконує в додатку. Модель MVC — це що повинні представити в коді, наприклад книги користувача, його банківського рахунку або чогось ще. Модель відповідає за зберігання функцій і змінних, які пов'язані з тим, що вона представляє. Можна думати про логікою моделі як базової концепції об'єктно-орієнтованого програмування. Тут моделі — це просто «класи». Не варто плутати з класами в контролерах, які технічно теж структуровані як класи. Нарешті, після того як контролер запитує інформацію з моделі, вона відправляє її подання. Вид схожий на систему шаблонів програми і може існувати для певного типу макета сторінки, мобільного виду або для певної теми. У виставі будуть

відобразитися всі розмітки і CSS, які традиційно використовуються під час створення статичної веб-сторінки.

MVC — це те, що бачить користувач, коли контролер звертається до нього. Контролер просто перенаправляє користувача на правильний вигляд, після того як вони отримали дані з моделі і перенаправили цю інформацію до подання. Потім подання відображає інформацію, надану їм, в тому форматі, в якому вона структурована.

Велика ідея MVC полягає в тому, що кожна частина коду має свою мету, і ці цілі різні. Деякі з кодів містять дані програми, деякі роблять додаток приємним, а деякі з них контролюють функціональність. Вважається, що це MVC, програма, яка здатна організувати основні функції коду в свої, акуратно організовані ящики. Структура файлу для використання MVC стандартним чином досить проста - є просто папки для уявлень, моделей і контролерів, і всі вони пов'язані один з одним через один каталог. Звичайно, з будь-яким веб-додатком у користувача також будуть інші папки та файли, такі як індексний файл і каталог для зображень.

Нижче представлена проста структура каталогів MVC з деякими прикладами файлів(рис.3.1).



Рисунок 3.1 — Структура каталогів MVC

JSON (JavaScript Object Notation) — простий формат обміну даними, що є зручним як для читання та написання людиною, так і для парсингу та генерації комп'ютером. Він базується на підмножині мови програмування JavaScript стандарту ECMA-262 3rd Edition - December 1999. JSON — це текстовий формат, повністю незалежний від мови реалізації, але він використовує конвенції, знайомі програмістам С-подібних мов, таких як С, С++, С#, Java, JavaScript, Perl, Python та багатьох інших. Ці властивості роблять JSON ідеальною мовою для обміну даними.

JSON базується на двох структурах даних:

Колекція пар ключ/значення. У різних мовах ця концепція реалізована як об'єкт, запис, структура, словник, хеш, іменованний список або асоціативний масив.

Упорядкований список значень. У більшості мов це реалізовано як масив, вектор, список або послідовність.

Це універсальні структури даних. У тому чи іншому вигляді їх підтримують майже усі сучасні мови програмування. Є сенс будувати формат даних, що є

незалежним від мови програмування, саме на цих структурах.

У нотації JSON це виглядає так:

Об'єкт — це невпорядкований набір пар ключ/значення. Об'єкт починається з {відкриваючої фігурної дужки і закінчується }закриваючою фігурною дужкою. Кожне ім'я супроводжують :двокрапкою, а пари ключ/значення відокремлюють , комою(рис.3.2).

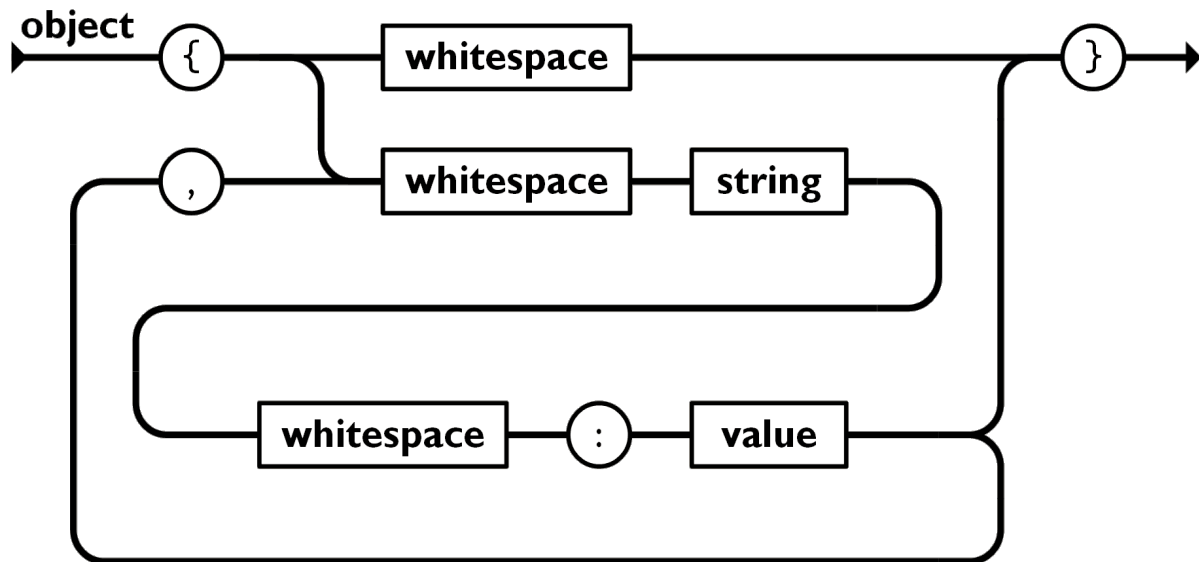


Рисунок 3.2 — Об'єкт у нотації JSON

Масив — це впорядкована колекція значень. Масив починається з [відкриваючої квадратної дужки і закінчується]закриваючою квадратною дужкою.

Значення відокремлюють ,комою(рис.3.3).

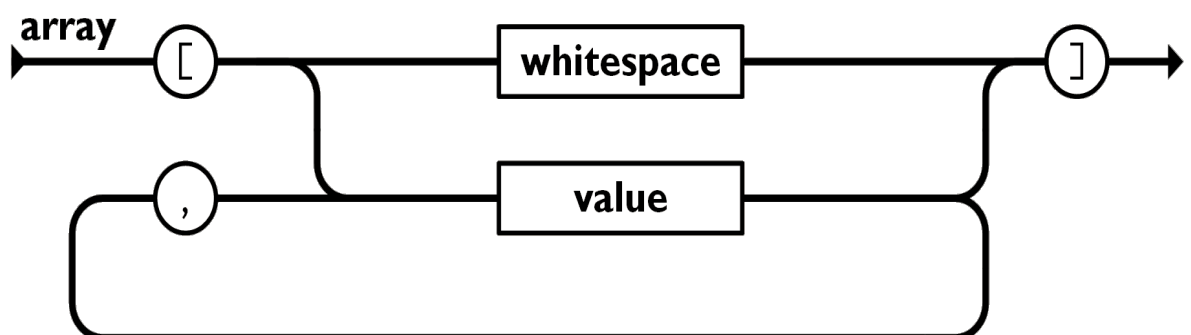


Рисунок 3.3 — Масив

Значення може бути строкою у подвійних лапках, числом, true, false, null,

об'єктом або масивом. Ці структури можуть бути вкладеними.

Строка — це послідовність з нуля або декількох символів Unicode, загорнута в подвійні лапки, що використовує `\` зворотною кошу риску задля екранування. Символ являє собою односимвольну строку. Строка у JSON дуже схожа до строк у C та Java.

Число зображується так само, як у C або Java, за виключенням того, що використовується тільки десятична система.

Пробіли можна використовувати поміж будь-якими лексемами.

За виключенням деяких подробиць кодування, це повністю описує мову.

Chart.js — це безкоштовна бібліотека JavaScript з відкритим вихідним кодом для візуалізації даних, яка підтримує вісім типів діаграм: стовпчаста, лінія, область, кругова, бульбашка, радар, полярна та розсіяна.

3.2 Архітектура проекту

Оскільки ми будемо використовувати технологію ReactJS, архітектура проекту буде дуже простою, а ReactJS js дозволить нам розгорнути базовий веб-додаток за допомогою однієї команди в консолі:

```
#npm create-react-app my-app
```

Замість відповіді Rest від сервера ми будемо використовувати файл json (Data.json), який містить дані про те, коли користувачі входять і виходять із сайту, приклад даних у цьому файлі наведено нижче:

```
{  
  "userId": "1",  
  "startDate": "2022-04-08",  
  "startTime": "00:01",  
  "endDate": "2022-04-08",  
  "endTime": "00:15"  
},
```

```
{  
  "userId": "2",  
  "startDate": "2022-04-08",  
  "startTime": "02:11",  
  "endDate": "2022-04-08",  
  "endTime": "02:15"  
},  
{  
  "userId": "3",  
  "startDate": "2022-04-08",  
  "startTime": "03:48",  
  "endDate": "2022-04-08",  
  "endTime": "03:51"  
}
```

Фактично, ми можемо отримати ці дані на сервер за допомогою REST-запиту, звичайно, ці дані спочатку потрібно зібрати та зберегти на сервері, але серверна частина не є показовою в контексті нашої роботи, логічної обробки даних користувача і кінцевий результат — відображення графіки з даними буде розроблено на стороні клієнта (front end), тобто на нашому веб-додатку.

Файлова структура проекту показана на рисунку 3.5:

Більш детально з кодом та структурою проекту можна ознайомитись переглянувши відкритий bitbucket репозиторій:

<https://bitbucket.org/v-protsenko/magistr-programm/src/master/>

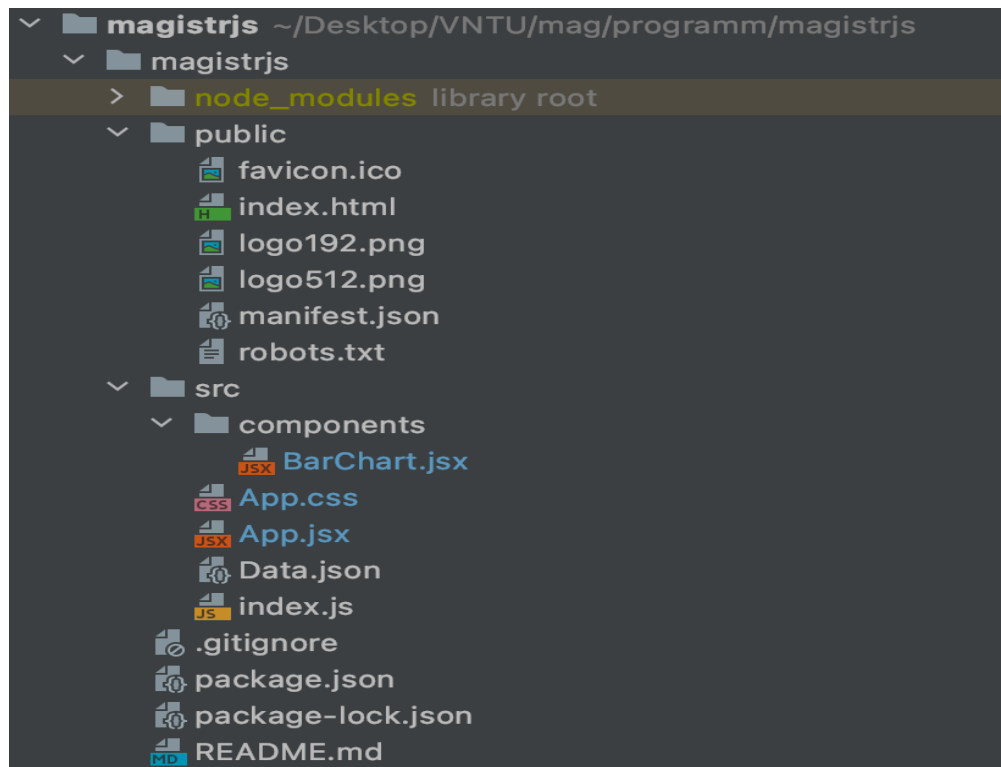


Рисунок 3.5 — Файлова структура проекту

3.3 Візуалізація роботи програмного рішення

Запустивши наш веб-додаток в середовищі розробки WebStorm за допомогою команди `npm start` у нас автоматично відкриється наш веб додаток в браузері по домену <http://localhost:3000/>

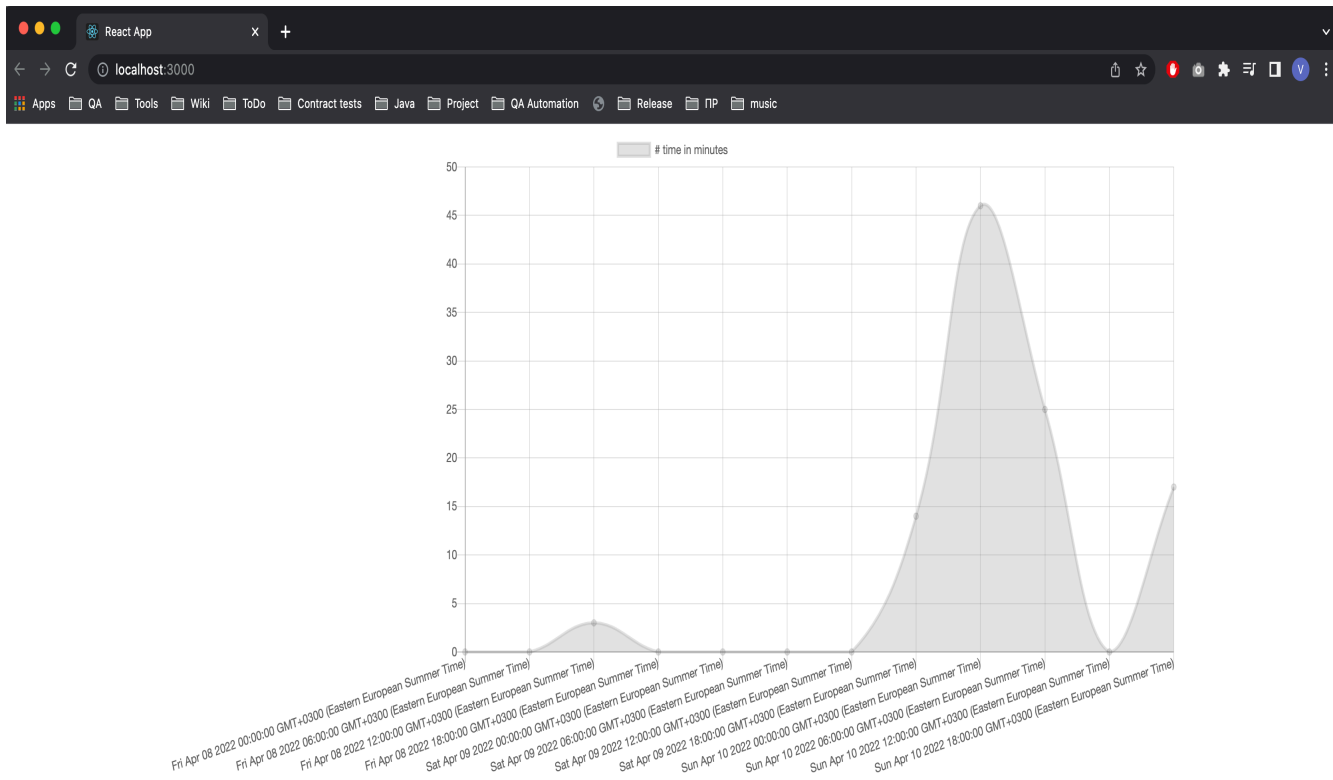


Рисунок 3.6 — Веб-додаток Magistrjs

Як же працює наш додаток, і які данні ми бачимо на цьому графіку(рис 3.6)? Для веб-аналітики, прогнозування наших KPI, планування різноманітних промоушенів зі скидками або рекламних кампаній, нам важливо відслідковувати у які періоди часу користувачі активно заходили до нас на сайт. Але самі по собі відвідування багато нам на скажуть про те, чи була рекламна кампанія дійсно вдалою, тому, що користувач міг перейти по рекламі, яка його зацікавила, але переглянувши сторінку на яку він попав, міг просто закрити сайт і більше на нього не повертатись.

Набагато більше нам розповість середній час перебування користувачів на сайті. Якщо ми бачимо, що за якийсь період користувачі дійсно проводили багато часу (більше ніж зазвичай) на нашому сайті (очевидно рекламна акція чи промоушн був вдалий), то ми можемо використати даний контент в майбутньому. Ну, і навпаки, якщо ми бачимо, що фактичний час перебування на сайті різко скоротився — очевидно є якісь проблеми, можливо навіть у нас не працюють окремі сторінки, чи

контент не подобається користувачам.

3.4 Вирішення задачі по обробці і графічному відображенню даних

Якщо абстрактно описати роботу нашого веб-додатку Magistrjs та розбити додаток на логічні частини, то перед нами постануть такі задачі.

Потрібно вчитати данні про користувачів з Data.json (отримати відповідь від сервера).

Данні про користувачів ми маємо, тепер по кожному конкретному користувачу нам потрібно вирахувати середній час перебування на сайті. Як бачимо (рис 3.7) рахувати різницю між даними виду нижче не дуже зручно.

```
"startDate": "2022-04-08",  
"startTime": "02:11",  
"endDate": "2022-04-08",  
"endTime": "02:15"
```

Тому для того, щоб порахувати час в хвилинах (середній час перебування на сайті) по конкретно взятому користувачу нам потрібно перевести дати з людино зрозумілої форми в TimeStamp (Timestamp (дослівно з англ. печатка/відмітка про час) також позначка часу — це послідовність символів або закодованої інформації, що показує, коли відбулася певна подія. Зазвичай показує дату і час (іноді з точністю до часток секунди).)

Різницю між TimeStamp вирахувати доволі просто, це ми робимо за допомогою цього блоку кода:

```
//PREPARE DEFINED VALUES
```

```
const definedValues = new Map();  
const definedKeys = [];
```

```

for(let i = 0; i < data.length; i++){
  let start = Date.parse(data[i].startDate + " " + data[i].startTime);
  let end = Date.parse(data[i].endDate + " " + data[i].endTime);
  let difference = end - start;
  let minutesDifference = Math.floor(difference/1000/60);
  definedKeys.push(start);
  definedValues.set(start, minutesDifference);
}
//PREPARE DEFINED VALUES
//CALCULATE RANGE BETWEEN FIRST AND LAST ITEMS AND ADD TO LIST
let allStamps = [];

let firstStamp = Date.parse(data[0].startDate + " " + "00:00")
let lastStamp = Date.parse(data.slice(-1)[0].startDate + " " + "23:59")

while (firstStamp < lastStamp){
  allStamps.push(firstStamp);
  firstStamp += 21600000;
}
//CALCULATE RANGE BETWEEN FIRST AND LAST ITEMS AND ADD TO LIST

```

Наступна задача, що стоїть перед нами, це підготувати коректні данні для відображення на нашому графіку. В чому тут може бути проблема? Ми вже маємо час перебування на сайті по кожному користувачу. Але проблема в тому що ми хочемо бачити данні за визначені проміжки часу, наприклад кожних 6 годин, тобто в 00:00, 06:00, 12:00, 18:00. Але користувачі не обов'язково перебували в цей час на нашому сайті. Якщо ж брати найближчі значення (наприклад користувач зайшов в 00:01), то такі данні також не будуть відображати реальну картину, так як один користувач міг перебувати на сайті 2 хвилини, а два інших 40 хвилин кожний. Як нам вирішити цю

задачу? Тут нам на допомогу прийде метод інтерполяції Лагранжа. Як це працює ?

Якщо описати цей метод абстрактно, то у нас є графік з осями X та Y , і є відомі нам точки між цими осями $X_i Y_i$. Метод Лагранжа дозволяє нам вирахувати криву, що з'єднує відомі нам точки, і отримати данні по $X_i Y_i$ на будь-якому відрізку цієї кривої(рис.3.7).

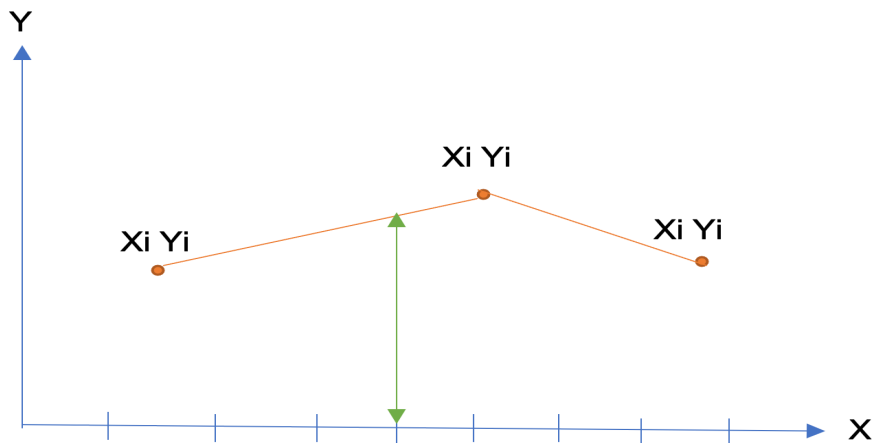


Рисунок 3.7 — метод інтерполяції Лагранжа

Приклад метода Лагранжа на JavaScript наведено нижче. `<script>`

```
// Javascript program for implementation
```

```
// of Lagrange's Interpolation
```

```
// To represent a data point
```

```
// corresponding to x and y = f(x)
```

```
class Data
```

```
{
```

```
  constructor(x,y)
```

```
  {
```

```
    this.x=x;
```

```
    this.y=y;
```

```

    }
}
// function to interpolate the given
// data points using Lagrange's formula
// xi corresponds to the new data point
// whose value is to be obtained n
// represents the number of known data points
function interpolate(f,xi,n)
{
    let result = 0; // Initialize result

    for (let i = 0; i < n; i++)
    {
        // Compute individual terms of above formula
        let term = f[i].y;
        for (let j = 0; j < n; j++)
        {
            if (j != i)
                term = term*(xi - f[j].x) / (f[i].x - f[j].x);
        }
        // Add current term to result
        result += term;
    }

    return result;
}

// Driver code

```

```
// creating an array of 4 known data points
let f=[new Data(0, 2), new Data(1, 3),
      new Data(2, 12), new Data(5, 147)];

// Using the interpolate function to obtain
// a data point corresponding to x=3
document.write("Value of f(3) is : " +
              interpolate(f, 3, 4));

// This code is contributed by rag2127
</script>
```

Ми ж змінимо його щоб, він працював в з нашими даними:

```
//LAGRANGE INTERPOLATION

const [dataMap, setDataMap] = useState([]);

class Data
{
  constructor(x,y)
  {
    this.x=x;
    this.y=y;
  }
}
```

```
function interpolate(f,xi,n)
{
  let result = 0; // Initialize result

  for (let i = 0; i < n; i++)
  {
    // Compute individual terms of above formula
    let term = f[i].y;
    for (let j = 0; j < n; j++)
    {
      if (j !== i)
        term = term*(xi - f[j].x) / (f[i].x - f[j].x);
    }

    // Add current term to result
    result += term;
  }

  return result;
}

let f= []
for(let x = 0; x < definedKeys.length; x++)
{
  f.push(new Data(definedKeys[x], definedValues.get(definedKeys[x])))
}

console.log("Value of f(3) is : " +
```

```
interpolate(f, allStamps[1], 2));
```

```
let allData = [];
```

```
for(let l = 0; l < allStamps.length; l++){
```

```
  allData.push(new Date(interpolate(f, allStamps[l], f.length)).getMinutes());
```

```
}
```

```
console.log(allData);
```

```
//LAGRANGE INTERPOLATION
```

Остання задача що стоїть перед нами, це відобразити данні на графіку. Для того, щоб створити сам графік ми будемо використовувати плагін react-chartjs в який ми передамо наші данні:

```
return (
```

```
<div>
```

```
<Line
```

```
  data={{
```

```
    labels: allDate,
```

```
    datasets: [{
```

```
      label: '# time in minutes',
```

```
      data: allData,
```

```
    ]
```

```
  }}
```

```
  height={600}
```

```
  width={400}
```

```
  options={{maintainAspectRatio: false}}
```

4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ

4.1 Порівняльний аналіз з Google Analytics

У ході проведеного тестування програмного продукту Magistrjs було перевірено відповідність програмного продукту поставленим вимогам. Підтверджено, що Magistrjs виконує наступні функції:

- обробляє данні про користувачів в форматі json;
- відображає середній час перебування користувачів на сайті в формі графіків(рис.4.1);
- розділяє графік на відповідні періоди для прикладу кожні 6 годин.

На підставі отриманих результатів було проведення дослідження ефективності розробленого продукту у порівнянні з існуючими технічними рішеннями, зокрема з Google Analytics.

Таким чином, було виявлено, що Google Analytics зберігає отримані дані використовуючи Cloud service, що несе ряд недоліків.

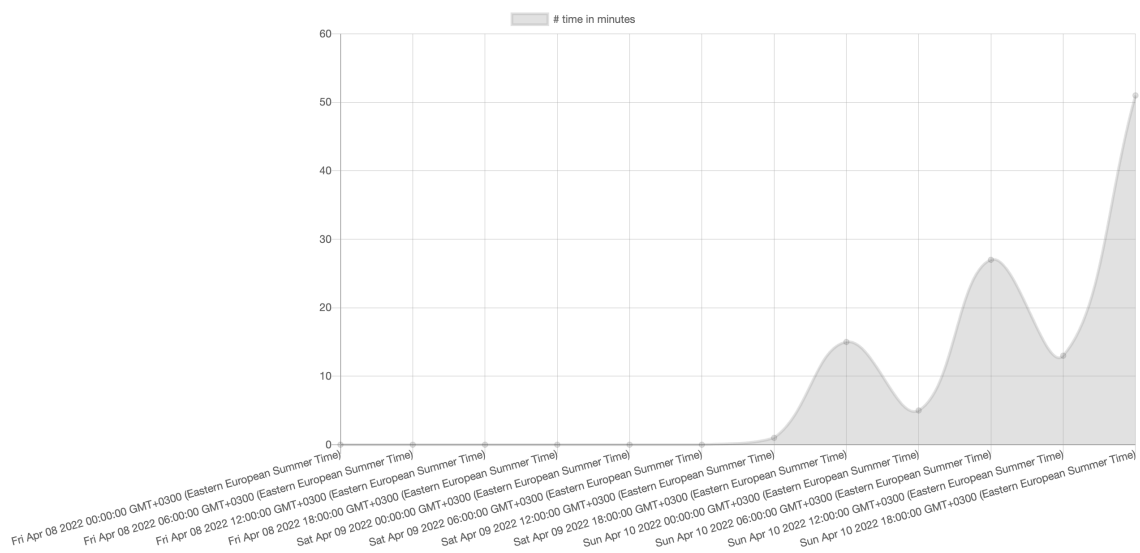


Рисунок 4.1 — середній час перебування користувачів на сайті

Перш за все, це не гарантує цілковиту конфіденційність та безпеку особистих

даних користувача від їх потрапляння до третіх осіб. На противагу такому рішенню в нашому додатку застосовується тип розгортання Standalone, що забезпечує безпеку даних для кожного окремого користувача. Крім того, на відміну від Google Analytics, наш додаток дозволяє розгортати власну інфраструктуру.

Наступною важливою перевагою Magistrjs є можливість безмежного зберігання та доступу до власних даних, у той час як Google Analytics дає таку можливість лише на 25 місяців.

Magistrjs дозволяє кожному користувачу проводити трекінг безмежної кількості параметрів, а Google Analytics лише певних 50.

При експорті даних Magistrjs забезпечує можливість експорту необмеженої кількості даних, а Google Analytics лише 5 000 за раз.

Можливість отримання з Magistrjs звітів без дискретизації, тобто первинних даних, які не були попередньо оброблені програмним продуктом. Google Analytics надає користувачеві лише звіти з попередньою обробкою, без доступу до отриманих програмою вхідних даних.

Ключовою перевагою Magistrjs над конкурентами є доступність отримання даних з усіх популярних браузерів. Це було доведено в ході тестування на Chrome, Mozilla, Opera, Safari, Edge. Google Analytics стикнулись з проблемою, що, наприклад, браузер Opera блокує дані для них з метою збереження конфіденційності дій користувачів. Крім того, розширення для браузера AdBlock також блокує передачу даних до Google Analytics, що в рази зменшує точність аналітики.

4.2 Визначення ключових переваг над конкурентами

Через велику кількість створених програмних рішень постає складність у детальному аналізі порівняння з усіма конкурентами. Тому в таблиці 4.1 наведено головні переваги успішних рішень та їх сьогодення вартість для користувача.

Таблиця 4.1 — Порівняння основних переваг з конкурентами

Програмний продукт	Головна перевага	Вартість
Magistrjs	Детальний аналіз користувачів	0 - 10\$ на одного користувача
Google analytics	Інтеграція з усіма продуктами Google	Безкоштовно (стандарт) 150 000\$ (преміум версія)
Clicky	Наявність мобільної версії	Безкоштовно
Mint	Відображення статистики в реальному часі	30\$ один веб-сайт
Woopra	Desktop додаток	0 - 1 999 \$
Mouseflow	Відео-аналітика дій користувача	13\$ в місяць
Crazy Egg	використовує технологію Heatmap	9\$ в місяць

4.3 Статистичні методи в А / В тестуванні

А / В тестування — одне з найбільш важливих завдань в веб-аналітиці. Щоб результати тестування стали по-справжньому корисною інформацією, якій можна довіряти, їх потрібно перевіряти на достовірність. Для цього у нагоді стане статистика. При проведенні А / В тестів слід враховувати такі поняття: статистична потужність, довжина вибірки, довірчий інтервал і статистична значимість.

Статистична потужність вимірюється у відсотках і визначає, наскільки ймовірно, що тест покаже різницю між варіантом А та В, якщо вона є. Чим більше довжина (розмір) вибірки для тесту, тим вище його статистична потужність.

Недостовірними є тести, статистична потужність яких нижче 80%.

Наступний аспект, який потрібно враховувати при А / В тестування — це статистична значимість. Вона визначає, наскільки ймовірно, що тест показав різницю між варіантом А та В, якої насправді не існує. Оптимальний рівень значимості для А/В тесту (також його називають довірчою ймовірністю) становить 95%. Тобто ймовірність помилки (так зване Р-значення) становить решта 5%. Статистична значимість тесту залежить від довірчих інтервалів і площі їх перетину.

Довірчий інтервал показує, наскільки стабільний показник, який ви отримали в результаті тесту, тобто повториться він, якщо збільшити вибірку. Наприклад, одній тисячі відвідувачів певного сайту показували картинку з зображенням тварин (варіант А), і 30% поставили їй лайк. Обчислюємо похибку для довірчого інтервалу за допомогою і отримуємо $\pm 2,8\%$. Це означає, що якщо збільшити вибірку, то з ймовірністю в 95% на лайк поставлять від 27,2% до 32,8% відвідувачів. Іншій тисячі відвідувачів показали зображення спортивного автомобіля (варіант В), і їй лайк поставили 26%. Довірчий інтервал для цієї групи — від 23,3% до 28,7%. Якщо порівняти довірчі інтервали для варіантів А і В, ми побачимо, що вони перетинаються в діапазоні від 27,2% до 28,7%, що відображено на рисунку 4.2.

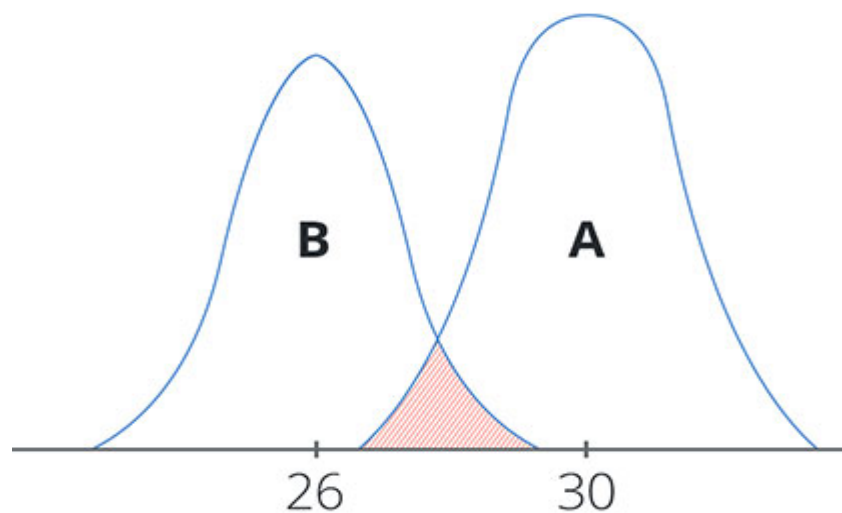


Рисунок 4.2 — Перетин довірчих інтервалів А та В

На горизонтальній осі графіка — КРІ (конверсія), на вертикальній осі — щільність розподілу випадкової величини. Чим менше площа перетину довірчих інтервалів, тим вище вірогідність результатів тестування. У нашому прикладі це перетин становить 1,5%. Ця цифра не перевищує Р-значення в 5%, а значить проведений тест є достовірним.

При вступі користувачів в комунікацію в певного структурі інтернет-ресурсу виділяється тематична інтернет-спільнота, де всім учасникам призначається унікальний ідентифікаційний номер. Таким чином, якщо розділити всю можливу активність користувачів по етапах, то надається можливим розробити спеціальний рейтинг активності користувачів інтернет-ресурсу. Розглянемо етапи активності аудиторії.

Етап 0 — пасивний стан (r_0), користувачі складаються в інтернет спільноті, але ніякої активності не проявляють, можливо повне ігнорування зовнішнього впливу на них.

Етап 1 — перегляд профілю (r_1), активність користувачів обмежується тільки переглядом головної сторінки інтернет-ресурсу підприємства.

Етап 2 — споживання контенту (r_2), користувачі здійснюють перегляд контенту інтернет-ресурсу, але не роблять більш складних дій.

Етап 3 — взаємодія з контентом (r_3), користувачі здійснюють перегляд контенту, а так само користуються функціями односторонньої комунікації (коментарі та позначки про збіг інтересів).

Етап 4 — вчинення комунікації (r_4), спілкування з представниками промислового підприємства або з іншими користувачами інтернет-ресурсу.

Етап 5 — вчинення комерційного дії (r_5).

Після виявляються всі дії кожного користувача і співвідносяться з сформованим раніше етапах активності. У підсумку, користувачеві призначається той етап активності, який охарактеризований наявністю максимального типу активності. Наприклад, якщо користувач споживав контент, мав комунікацію з представниками

підприємства і зробив покупку продукції, то йому призначається 5 етап активності. Таким чином, сума часток етапів 4 (r_4) та 5 (r_5) буде характеризуватися як загальний показник активності аудиторії інтернет-ресурсу, що можна представити в математичному вигляді:

$$K_{\text{взаимод.}} = \frac{r_4 + r_5}{q_{\text{рес.}}},$$

де $K_{\text{взаимод.}}$ — коефіцієнт взаємодії аудиторії інтернет-ресурсу;

q — загальна кількість користувачів інтернет-ресурсу;

r_4 — частка аудиторії по 4 етапу активності інтернет-ресурсу;

r_5 — частка аудиторії по 5 етапу активності інтернет-ресурсу.

Підсумковий показник активності аудиторії являє собою коефіцієнт, який об'єктивно показує наскільки успішно йде залучення користувачів в вивчення контенту інтернет-ресурсу і в комунікацію з його представниками.

Математична модель використана для реалізації проекту ґрунтується методах статистики. За її допомогою ми визначаємо чи виходить отримана величина за межі області допустимих значень даної метрики (так званий довірчий інтервал), чи являється зміна критичною

Створене програмне рішення дозволяє визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки про пріоритетність у соціальному житті населення (в т.ч. по регіонах), а для комерційних представників — розробляти бізнес-стратегії для просування сайтів та охоплення нової аудиторії, базуючись безпосередньо на числових даних отриманих у ході дослідження.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Опис ідеї проекту

В економічній частині кваліфікаційної роботи досліджується обґрунтування економічної доцільності проведеної роботи, зокрема — розробка та реалізація системи аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з мережі Інтернет.

В роботі досліджується поетапний процес розробки такої моделі, а також наведено оцінювання отриманих даних .

Запропонована структурна схема технології, з точки зору якості процесу розробки даної моделі, має ряд переваг для користувачів в різних напрямках застосування.

В таблиці 5.1 наведено цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати потрібні групи потенційних клієнтів.

Таблиця 5.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка та реалізація системи аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з мережі Інтернет.	Для веб-маркетологів	Визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу
	Для комерційних представників	Розробляти бізнес-стратегії для просування сайтів та охоплення нової аудиторії
	Для соціологів	Робити висновки про пріоритетність у соціальному житті населення (в т.ч. по регіонах)

На підставі проведеного аналізу результатів така методика дозволяє поліпшити робочий процес без суттєвих втрат якості моделі.

Метою даного розділу є економічне обґрунтування витрат замовника щодо проведення науково-дослідних робіт з дослідження вказаних питань.

Основою розрахунків буде визначення трудовитрат та заробітної плати працівникам, одноразових витрат, прибутку, оцінка економічної ефективності.

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів наведено в таблиці 5.2.

Таблиця 5.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Потенційні концепції конкурентів			W (слабка сторона)	N нейтральна сторона)	S сильна сторона
		мій	Google Analytics	Piwik			
1	Візуалізація обсягу трафіку і поведінкових показників по днях	+	+	+		+	
2	Надання класифікації відвідувачів у формі звіту	+	+	-			+
3	Аналіз цінності нового контенту	+	+	+			+

Як видно з таблиці 5.2 головною перевагою проекту для користувачів являється збільшення точності вимірювання та зменшення часу на проведення аналізу. Не менш важливим, є такі техніко-економічні характеристики, як ціна та простота алгоритму. Слабку позицію займає простота програми, так як для виконання апроксимації складних функцій необхідний більш складний алгоритм.

5.2 Технологічний аудит ідеї проекту

У таблиці 5.3 проводиться аудит технології, за допомогою якої можна реалізувати ідею проекту.

Таким чином, технологічна реалізація проекту є цілком можливою, з поміж названих технологій доцільно використати усі перераховані, адже кожна з них є доступною та наявною на ринку. Технологічний шлях передбачає розробку інструменту для веб-аналітики, який дає змогу відстежити обсяг трафіку і поведінкові показники по днях, а усі отримані дані візуалізувати на клієнтському інтерфейсі.

Таблиця 5.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Візуалізувати обсяг трафіку і поведінкові показники по днях	RabbitMQ, PHP	+	+
2	Збір даних про користувача	JavaScript, BeaconAPI	+	+
3	Відобразити сегментацію відвідувачів в формі діаграм	PHP, PowerBI	+	+
4	Надавати зміни поведінкових показників		необхідна розробка	+
5	Аналізувати цінність нового контенту	PHP, RabbitMQ	+	+

5.3 Аналіз ринкових можливостей запуску проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів наведено у таблиці 5.4.

Таблиця 5.4 — Попередня характеристика потенційного ринку проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	3 (Google Analytics, Piwik, Clicky)
2	Загальний обсяг продаж, грн/ум.од	15000 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Велика конкуренція
5	Специфічні вимоги до стандартизації та сертифікації	_____

Отже, за результатами аналізу таблиці 5.4 та попереднім оцінюванням сьогодні ринок є доволі привабливим для входження за умов подолання бар'єру високої конкуренції.

Для отримання більш чіткої картини ситуації на ринку визначимо потенційні групи клієнтів (табл.5.5), їх характеристики, сформуємо орієнтовний перелік вимог до товару для кожної групи.

Таблиця 5.5 — Характеристика потенційних клієнтів проекту

Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Аналіз поведінки людини в веб-середовищі	Веб-маркетологи та власники веб-сайтів	Дотримання плану продажів продукції	Отримання класифікації відвідувачів у формі звіту
	Соціологи	Регіонарні або державні соціологічні дослідження	Звітність про пріоритетність у соціальному житті населення (в т.ч. по регіонах)
Аналіз цінності нового контенту	Власники компаній та SEO	Унікальність контенту, його сприйнятливості користувачем, кількість відгуків	Спрощення отримання даних для розробки бізнес-стратегії для просування сайтів та охоплення нової аудиторії

Проведемо аналіз факторів ринкового середовища що сприяють ринковому впровадженню (табл. 5.6.) проекту, та факторів, що йому перешкоджають (табл. 5.7.). Результати наведено у таблицях 4.6 — 4.7.

Такі системи використовуються в досить нових та прогресивних сферах, тому попит на них постійно зростає, тим самим збільшуючи ринок клієнтів.

Таблиця 5.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Наявність як мінімум трьох серйозних конкурентів на ринку (Google Analytics, Piwik, Clicky)	Порівняння їх основних особливостей, вибір варіанту з більшою кількістю переваг
2	Високий бар'єр входження в технологію	Складність у вивченні технології та всіх пов'язаних з аналітикою особливостей - веде до збільшення фінансових витрат	Відмова від використання технології
3	Час на налаштування всіх функціональних можливостей програмного продукту та потреба у персональному спеціалізованому налаштуванні під конкретні потреби деяких клієнтів	Збільшення часу на розробку та налаштування, а також можливе некоректне відображення даних у перші місяці після запуску	Звернення у службу підтримки продукту

Збільшення попиту на товар — призведе до різкого збільшення об'єму виробництва та продажу, що дасть поштовх до нових вдосконалень. Це досягається шляхом реклами та освоєння нових сфер використання.

Головним фактором загроз є конкуренція. Вже існуючі товари на ринку мають певну репутацію та великі об'єми виробництва.

Таблиця 5.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Аналіз поведінки людини в веб-середовищі	Візуалізувати обсяг трафіку і поведінкові показники по днях	Стрімке визначення необхідного аналітичного інструменту
2	Визначення “портрету” цільового клієнта	Виводити класифікацію відвідувачів у формі звіту	Збільшення доходів компанії та проведення змін в стратегії залучення нових клієнтів
3	Аналіз цінності нового контенту	Одномоментне визначення унікальності контенту, його сприйнятливості користувачем та кількості відгуків	Внесення коректив до бізнес-стратегії для просування сайтів

Конкуренти здатні демпінгувати ціни для отримання нових клієнтів свого товару.

Надалі проведемо аналіз пропозиції (табл.5.8.), де визначимо загальні риси конкуренції на ринку.

Таблиця 5.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Тип конкуренції: чиста	На ринку представлено велику кількість продавців і покупців, пропозиція однакових товарів та послуг для продажу	Можливість вільної появи на ринку та виходу з нього покупців та продавців.

Продовження таблиці 5.8

За рівнем конкурентної боротьби: міжнародна	Вид конкурентної боротьби виробників на світовому ринку, яка поєднує внутрішньогалузеву і міжгалузеву конкуренцію.	Така конкуренція сприяє збалансованому розвитку світового ринку, переливу капіталу не тільки між сферами виробництва, а й між різними країнами
За галузевою ознакою: внутрішньогалузева	Це конкуренція між товаровиробниками, що діють в одній галузі народного господарства.	Характеризується зниженням витрат на виробництво й стимулює процес концентрації виробництва та капіталу.
Конкуренція за видами товарів: між бажаннями	Бажання, які споживач може прагнути задовольнити	Розуміння прагнень та актуальних потреб клієнта

На ринку присутня чиста конкуренція через те, що окремі гравці не можуть впливати на ціну товару. За рівнем конкурентної боротьби – міжнародна з міжгалузевою ознакою, так як можливі іноземні замовники. Конкуренція за видами товарів – видова. Вартість товару напряму залежить від комплектації та функціоналу, тому за характером конкурентних переваг – цінова.

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі за моделлю п'яти сил М. Портера (табл. 5.9.)

Таблиця 5.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти	Потенційні конкуренти	Постачальники	Клієнти
	Google Analytics, Piwik, Clicky	24Log	Google	Веб-маркетологи, комерційні представники, соціологи
Висновки:	Відомі технологічні рішення, що використовуються для аналізу поведінки людей у веб-середовищі	Надає три варіанти лічильника: з детальною статистикою про відвідувачів, лічильник "скільки людей на сайті" і простий лічильник відвідуваності	Сервіс, що надається Google для створення детальної статистики відвідувачів веб-сайтів. Статистика збирається на сервері Google, користувач тільки розміщує JS-код на сторінках свого сайту	Бажання спрощено отримувати дані для розробки бізнес-стратегії для просування сайтів та охоплення нової аудиторії

На основі аналізу конкуренції, проведеного наведеного в табл. 5.9, а також із урахуванням характеристик ідеї проекту, що наведені в табл. 5.2, вимог споживачів до товару (табл. 5.5.) та факторів маркетингового середовища (табл. 5.6, 5.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Аналіз наведено в таблиці 5.10.

Таблиця 5.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Візуалізація обсягу трафіку і поведінкові показники по днях	Аналіз поведінки людини в веб-середовищі
2	Надання класифікації відвідувачів у формі звіту	Визначення “портрету” цільового клієнта
3	Аналіз цінності нового контенту	Одномоментне визначення унікальності контенту, його сприйнятливості користувачем та кількості відгуків
4	Дешевизна програмного продукту	Зацікавленість роботодавців в економії бюджету
5	Зростаюча динаміка ринку	Змога вільно увійти на ринок та мати клієнтів

З огляду на конкурентну ситуацію, можна зробити висновок, що загрозу становлять три основні конкуренти: Google Analytics, Piwik, Clicky, а також потенційний конкурент 24Log, однак зростаюча динаміка ринку та чиста конкуренція на ньому дають змогу вільно увійти на ринок.

За визначеними факторами конкурентоспроможності (табл. 5.10) проведемо аналіз сильних та слабких сторін стартап-проекту (табл. 5.11).

Таблиця 5.11 — Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів							
			-3	-2	-1	0	+1	+2	+3	
1	Візуалізація обсягу трафіку і поведінкові показники по днях	20								+
2	Надання класифікації відвідувачів у формі звіту	20							+	
3	Аналіз цінності нового контенту	15				+				
4	Дешевизна програмного продукту	10					+			
5	Зростаюча динаміка ринку	15								+

Підсумовуючи рейтинг товару відносно головного конкурента, запропонована система має більший рейтинг відносно прямих конкурентів. Дана таблиця показує якими саме особливостями розроблена система відрізняються від аналогів та в яку саме сторону. Детальний аналіз показує, що сильними сторонами є простота, можливість до адаптації з роботою із іншими системами.

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities), що наведено в таблиці 5.12. на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін в таблиці 5.11.

Таблиця 5.12 — SWOT аналіз проекту

Сильні сторони:	Слабкі сторони:
<ul style="list-style-type: none"> - Візуалізація обсягу трафіку і поведінкові показники по днях; - Надання класифікації відвідувачів у формі звіту; - Аналіз цінності нового контенту; 	<ul style="list-style-type: none"> - Час на налаштування всіх функціональних можливостей програмного продукту; - Можливі помилки у програмі;
Можливості:	Загрози:
<ul style="list-style-type: none"> - Зростаюча динаміка ринку; - Чистий тип конкуренції в галузі. 	<ul style="list-style-type: none"> - Високий бар'єр входження в технологію; - Наявність як мінімум трьох серйозних конкурентів на ринку (Google Analytics, Piwik, Clicky).

На основі SWOT аналізу розробимо альтернативи ринкової поведінки (перелік заходів) для виведення проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Розробка ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл.5.13).

Таблиця 5.13 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Веб-маркетологи та SEO	Висока	Високий	Висока	Середня
2	SEO	Висока	Високий	Середня	Середня

Продовження таблиці 5.13

3	Соціологи	Середня	Низький	Середня	Середня
Обрано цільову групу: Веб-маркетологи, CEO та SEO.					

За результатами аналізу потенційних груп споживачів ми обрали ті, які серед усіх інших груп, існуючих у даному суспільному середовищі, найбільше потерпають від цієї проблеми, на розв'язання якої переважно спрямований проект та групи які можуть найбільше вплинути на характеристики цієї проблеми — зміст, структуру, масштабність, характер, соціальну значимість тощо.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 5.14.).

Таблиця 5.14 — Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
Фланкова атака	Диференційованого маркетингу	Надання класифікації відвідувачів у формі звіту та аналіз цінності нового контенту	Стратегія диференціації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 5.15).

Таблиця 5.15 — Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні, наявність як мінімум трьох серйозних конкурентів на ринку	Намагатиметься залучити нових та повернути увагу клієнтів, що користуються рішенням конкурентів	В основу покладено візуалізацію обсягу трафіку і поведінкові показники користувачів	Стратегія виклику лідера

Оскільки проект не є першопрохідцем та має суттєві переваги по відношенню до свого прямого конкурента, можливо обрати стратегію виклику лідера. Це є можливим на фоні використання інноваційних методів підвищення точності алгоритмічного забезпечення, впровадження інновацій і технологій для вдосконалення продукту з метою ускладнення задачі конкурентів

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування (табл. 5.16).

Таблиця 5.16 — Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту
Надання класифікації відвідувачів у формі звіту	Стратегія диференціації	Візуалізація обсягу трафіку і поведінкові показники по днях; Надання класифікації відвідувачів у формі звіту; Аналіз цінності нового контенту;	аналізує контент; класифікує відвідувачів дешевий;

Обравши конкретний сегмент, фірма має визначити для себе, як проникнути на нього й здобути конкурентні переваги. Перед цим потрібно дослідити позиції всіх наявних на ринку конкурентів і визначити своє місце на ринку. На ряду з використанням інноваційних методів підвищення точності, проект повинен викликати асоціації у клієнта у гнучкості налаштування та комплексування, високої якості виготовлення.

З огляду на конкурентну ситуацію, можна зробити висновок, що загрозу становлять три основні конкуренти: Google Analytics, Piwik, Clicky, а також потенційний конкурент 24Log, однак зростаюча динаміка ринку та чиста конкуренція на ньому дають змогу вільно увійти на ринок.

Обрано стратегію диференційованого маркетингу для охоплення ринку, що орієнтована на веб-маркетологів, SEO та SEO.

Зважаючи на поточне конкурентне становище обрано стратегію виклику лідера з альтернативною фланговою атакою, яка передбачає атаку на слабкі сторони фірми-лідера.

5.4 Розрахунок витрат на здійснення науково-технічної розробки

Для проведення аналізу якості розробленої методики створеної моделі шляхом порівняння її з моделями конкурентів на підставі короткого опитування було залучено до роботи 10 експертів, на умовах погодинної оплати. Загальний обсяг часу, відведеного для опитування, складав 5 годин на експерта, з оплатою 120 грн/годину. Також було запрошено спеціаліста-аналітика в сфері дизайну інтерактивних видань, заробітна плата якого становить 12000 грн/міс. Заробітна плата керівника робіт становить 15000 грн/міс, заробітна плата основного виконавця робіт визначена в 12000 грн/міс.

Проведемо розрахунок трудовитрат і заробітної плати виконавців робіт. Середньоденна заробітна плата виконавця робіт ($Z_{\text{ср.дн.}}$) розраховується:

$$Z_{\text{ср.дн.}} = \frac{Z_{\text{ср.міс.}}}{n}, \quad (5.1)$$

де $Z_{\text{ср.міс.}}$ — середньомісячна зарплата виконавця роботи;

n — число робочих днів у місяці, ($n = 22$).

В даному випадку середньоденна заробітна плата керівника проекту складає 681,82 грн/день, заробітна плата аналітика та виконавця складає по 545,45 грн/день.

Етапи виконання НДР, перелік і зміст робіт, трудомісткість їх виконання, заробітна плата виконавця робіт представлені в табл. 5.17.

Таблиця 5.17 — Розрахунок трудовитрат і заробітної плати виконавця робіт

Перелік робіт	Кількість виконавців	Трудомісткість робіт, люд.-днів	Середньоденна заробітна плата, грн.	Сума заробітної плати, грн.
1. Підготовчий етап				
1.1. Розробка та затвердження ТЗ	1	1	681,82	681,82

Продовження таблиці 5.17

1.2 Підготовка аналітичних матеріалів та даних для виконання НДР	1	4	545,45	2181,82
2. Основний етап				
2.1 Постановка задачі	1	1	681,82	681,82
Етап 1. Огляд сучасних методів даних в комп'ютерних системах.	1	1	545,45	545,45
Етап 2. Збір даних з веб-середовища для розробки системи веб-аналітики.	1	3	545,45	1636,36
Етап 3. Вибір засобів розробки	1	1	545,45	545,45
Етап 4. Розробка програмного та технічного забезпечення	1	4	545,45	2181,82
2.2 Практичний експеримент, перевірка розробленої методики	1	4	545,45	2181,82
2.3 Проведення А/В тестування	10	5 год	120 грн/год	6000,00
3. Заключний етап				
3.1 Аналіз результатів проведення роботи	2	3	681,82	2045,45
			545,45	1000,00
3.2 Формування висновків та пропозицій за темою проекту	2	2	681,82	1363,64
			545,45	1000,00
Усього		24		22045,45

Калькуляція собівартості розраховується відповідно до існуючих нормативних актів України. До складу калькуляції входять такі статті витрат:

- матеріальні витрати;
- витрати на оплату праці;
- єдиний соціальний внесок;
- амортизація основних засобів (вартість машинного часу);
- витрати на спожиту електроенергію;
- інші витрати.

До інших витрат відносяться адміністративні витрати (водопостачання, водовідведення, опалення, освітлення) та вартість послуг зв'язку.

Матеріальні витрати визначаються витратами на матеріали, визначені їх потребою для виконання робіт, і цін, що діють на момент складання калькуляції. Матеріальні витрати розраховуються за такою формулою:

$$M = \sum_{j=1}^n Q_j \times C_j, \quad (5.2)$$

де M — сумарні витрати на матеріали, в тому числі малоцінні предмети, що швидко зношуються (носії, папір, канцелярське приладдя тощо), або на літературу, яка необхідна для проведення роботи, тощо;

Q_j — кількість використаних одиниць j -го виду матеріалів, $j=(1..n)$;

C_j — ціна одиниці j -го виду матеріалів.

Розрахунок матеріальних витрат представлено в табл. 5.18.

Витрати на оплату праці розраховуються виходячи з необхідного для виконання робіт складу й кількості працівників, а також із середньомісячної заробітної плати. Відповідно до проведених розрахунків витрати на оплату праці виконавців роботи дорівнюють 22045,45 грн.

Таблиця 5.18 — Розрахунок матеріальних витрат

Найменування	Од. вим.	Кількість, (Q_j)	Ціна (C_j), грн	Сумарні витрати на матеріали (M), грн
Олівець механічний	шт.	2	23,00	46,00
Блокнот	шт.	2	40,00	80,00
Усього				126,00

Єдиний внесок на загальнодержавне соціальне страхування (ЄСВ) — консолідований страховий внесок, збір якого здійснюється в систему загальнообов'язкового державного соціального страхування в обов'язковому порядку і на регулярній основі з метою забезпечення захисту у випадках, передбачених законодавством, прав застрахованих осіб і членів їх сімей на отримання страхових виплат (послуг) за діючими видами загальнообов'язкового державного соціального страхування.

Для об'єкта дослідження ставка єдиного соціального внеску дорівнює 22% від витрат на оплату праці, тобто розмір ЄСВ дорівнює 4 850,00 грн.

При виконанні НДР застосовувалось наступне обладнання: комп'ютер вартістю 21000 грн.

Вищенаведене устаткування є власністю організації виконавця, тому доцільно розрахувати суму амортизаційних відрахувань на період виконання НДР. Амортизація основних засобів розраховується за формулою:

$$A_{обл} = \frac{Ц_б}{T_e} \cdot \frac{t_{вик}}{12} \quad (5.3)$$

де $Ц_б$ — балансова вартість комп'ютера, який використовувався для проведення досліджень, грн;

$t_{вик}$ — термін використання комп'ютера під час досліджень, місяців;

T_e — строк корисного використання комп'ютера, років.

Підставивши відомі значення у (5.3), визначимо величину амортизаційних відрахувань:

$$A_{обл} = \frac{21000}{5} \cdot \frac{9}{12} = 3150,00 \text{ (грн.)}$$

Витрати на використану обладнанням електроенергію розраховуються за формулою:

$$З_e = M \cdot t \cdot T_{кВт}, \quad (5.4)$$

де M — потужність устаткування, тобто кількість енергії, споживаної за одиницю часу (кВт/година);

t — кількість годин використання устаткування за період проведення науково-дослідницької роботи;

$T_{кВт}$ — тариф, тобто вартість використання 1 кВт електроенергії.

Споживна потужність комп'ютера складає 0,5 кВт за годину, тариф

споживачів складає 0,9 грн./кВтгодин (без ПДВ). Підставивши значення у формулу (4.4), визначимо величину витрат на спожиту електроенергію:

$$Z_9 = 0,5 \cdot 128 \cdot 0,9 + 0,5 \cdot 64 \cdot 0,9 + 0,5 \cdot 8 \cdot 0,9 = 90 \text{ грн.}$$

До інших статей витрат відносяться такі:

— адміністративні витрати: (водопостачання, водовідведення, освітлення, опалення), які прийнято у розмірі 20% від витрат на оплату праці;

— вартість оплати послуг зв'язку.

Вартість оплати послуг зв'язку становитиме:

Інтернет — із розрахунку 250 грн. на місяць (безлімітний пакет); всього 250 грн. за 25 днів виконання НДР.

За час виконання НДР витрати на відрядження, аутсорсинг, інформаційні послуги та маркетингові заходи не мали місця.

Результати розрахунку кошторису витрат, тобто одноразових витрат, на виконання НДР наведені в табл. 5.19.

Таблиця 5.19 — Кошторис витрат на розробку НДР

№ з/п	Стаття витрат	Сума, грн.
1	Заробітна плата	22 045,45
2	Єдиний соціальний внесок (22,0 % від п.1)	4 850,00
3	Матеріальні витрати	126,00
4	Амортизація основних засобів	3150,00
5	Витрати на спожиту електроенергію	90,00
6	Інші витрати, у тому числі:	
6.1	адміністративні витрати (20% від п.1)	4 409,09
6.2	вартість послуг зв'язку	250,00
	Усього витрати на розробку (Вр)	34 920,54

Таким чином, кошторис витрат на виконання даної МКР складає 34920,54 грн.

5.5 Оцінка результатів науково-дослідної роботи

Результат — це наслідок послідовності дій виконаних при МКР, виражений

якісно або кількісно. В загальному випадку оцінка результатів МКР — це визначення ефективності отриманих рішень порівняно з сучасним науково-технічним рівнем.

$$\Delta P_j = |X_{бj} - X_{нj}|, \quad (5.5)$$

де ΔP_j — покращення j -ої характеристики досліджуваного процесу за рахунок впровадження результатів НДР ($j=1, m$);

m — кількість досліджуваних характеристик;

$X_{бj}$ — базове значення j -ої характеристики;

$X_{нj}$ — нове значення j -ої характеристики після впровадження НДР.

Таблиця 5.20 — Відсоток досягнення мети користувачем та оцінка якості дизайну

Характеристика	До впровадження	Після впровадження
Час розрахунку середнього часу перебування користувачів на сайті, хв	1,3	1

Підставивши відповідні значення до формули (5.20), визначимо результат від впровадження НДР у чисельному вигляді:

$$\Delta P_{ч.1} = |1,3 - 1| / 1,3 = 23,1\%;$$

За результатами дослідження покращення є вдалим. Вони зменшили витратний час на 23,1% та отримали задовільну оцінку за якість моделі.

Під час економічного обґрунтування МКР було дана характеристика науково-дослідної роботи, наведені етапи виконання МКР, розрахована їх трудомісткість та заробітна плата виконавця. Розраховано одноразові витрати на розробку МКР, дано оцінку результатів науково-дослідної роботи та визначено економічну ефективність результатів МКР. Дана науково-дослідна робота має позитивний показник економічної ефективності. Роботу у цілому можна враховувати ефективною або такою, що має науковий та технічний рівень.

ВИСНОВКИ

Використання вдосконалених алгоритмів апроксимації даних при розробці алгоритмічного та програмного забезпечення аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з мережі Інтернет за допомогою попереднього використання алгоритмів інтерполяції є важливим і актуальним завданням, що допомагає комерційним компаніям коригувати свої веб-сайти та бізнес-цілі, виявляючи області для вдосконалення, знаходження нових інструментів онлайн-просування сайту, тестування нових функціональних можливостей та, зрештою, збільшення власних прибутків.

1 Встановлено, що використання методів апроксимації даних при аналізі процесів взаємодії користувачів з веб-ресурсом надає нові можливості для веб-аналітики та сприяє вірному коригуванню веб-сайту та бізнес-цілей, виявляючи області для вдосконалення, знаходження нових інструментів онлайн-просування сайту, тестування нових функціональних можливостей та, зрештою, збільшення власних прибутків.

Математична модель ґрунтується на методі апроксимації даних та інтервальному методу інтерполяції за формулою Лагранжа.

2 Найбільш ефективним методом апроксимації даних отримуваних з користувацьких запитів під час використання веб-сайту є інтервальний метод інтерполяції Лагранжа, оскільки він надає гарні результати при глобальній інтерполяції вибірок даних, що характеризуються різним кроком зміни значень аргументу.

3 Для забезпечення інтерполяції отримуваних з веб-середовища даних за методом інтерполяційних поліномів Лагранжа було розроблено блок-схем алгоритму його реалізації та веб додаток на react js у середовищі розробки Web Storm. Під час розробки проекту Magistrjs була використана технологія ReactJS, що дозволило спростити архітектуру проекту. Замість Rest відповіді від сервера, було використано json файл (Data.json) з даними про час входу та виходу користувача на сайт. Сервіси

діляться на API сервіс, мікрос-служба аналізу даних, сервіс для керування користувачем.

Створене програмне рішення дозволяє визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки про пріоритетність у соціальному житті населення (в т.ч. по регіонах), а для комерційних представників — розробляти бізнес-стратегії для просування сайтів та охоплення нової аудиторії, базуючись безпосередньо на числових даних отриманих у ході дослідження.

Результати порівняння ефективності розробленого програмного продукту Magistrjs з головним конкурентом Google Analytics, показали, що новий продукт дозволяє більш точно визначати кількісні показники:

- Magistrjs дозволяє кожному користувачу проводити трекінг безмежної кількості параметрів, а Google Analytics лише певних 50;
- при експорті даних Magistrjs забезпечує можливість експорту необмеженої кількості даних, а Google Analytics лише 5 000 за раз.

У результаті розрахунку комерційного потенціалу нової розробки був отриманий позитивний показник економічної ефективності, що робить її комерційно привабливою для просування на ринку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Антамошкин О. А., Кукарцев В. В. Модели и методы формирования надежных структур информационных систем обработки информации // Информационные технологии и математическое моделирование в экономике, технике, экологии, образовании, педагогике и торговле. 2014. № 7. С. 51–94.
2. Толстова Ю.Н. Измерение в социологии: Курс лекций. М.: ИНФРА-М, 1998.
3. Hu, X., & Cercone, N. (2004). A Data Warehouse/Online Analytic Processing Framework for Web Usage Mining and Business Intelligence Reporting. *International Journal of Intelligent Systems*, 19(7), p. 585–606.
4. Рыбалко В. В. Параметрическое диагностирование энергетических объектов на основе факторного анализа в среде Statistica // Exponenta Pro.— 2004. — N 2.— с. 78-83.
5. Акобир Шахиди. Деревья решений – общие принципы работы // [Электронный ресурс] <http://www.basegroup.ru/trees/>
6. [Электронный ресурс] [http://ru.wikipedia.org/wiki/Корреляционный анализ](http://ru.wikipedia.org/wiki/Корреляционный_анализ)
7. Баскаков С.И. Радиотехнические цепи и сигналы / С.И. Баскаков. – М.: Высшая школа, 2000. – 446 с.
8. Попов В.П. Основы теории цепей / В.П. Попов.— М.: Высшая школа, 1998. – 575 с.
9. Радиотехнические цепи и сигналы. Задачи и задания / под ред. Я.Н.Яковлева. – М.: ИНФРА-М; Новосибирск: Изд-во НГТУ, 2003.
10. Иванов М.Т. Теоретические основы радиотехники / М.Т. Иванов, А.Б. Сергиенко, В.Н. Ушаков; под ред. В.Н. Ушакова. – М.: Высшая школа, 2002.
11. Аппроксимация [Электронный ресурс]. – Доступ: <https://studfiles.net/preview/380226/>
12. Ахиезер Н.И. Лекции по теории аппроксимации / Н.И. Ахиезер. – М.: Наука. Гл. ред. физ.-мат. лит., 1965.
13. Медведев В.С., Потёмкин В.Г. Нейронные сети. MATLAB 6 / Под

ред. В.Г.Потёмкина. – М.: Диалог-МИФИ, 2002. – 496 с

14. Введение в методы интерполяции [Электронный ресурс]. – Доступ: <https://desktop.arcgis.com/ru/arcmap/10.4/extensions/geostatistical-analyst/anintroduction-to-interpolation-methods.htm>

15. Интерполяційний многочлен Лагранжа [Электронный ресурс]. – Доступ: https://uk.wikipedia.org/wiki/Многочлен_Лагранжа

16. Интерполяційні формули Лагранжа для нерівновіддалених вузлів інтерполяції [Электронный ресурс]. – Доступ: <http://www.mathros.net.ua/interpoljacijni-formuly-lagranzha-dlja-nerivnoviddalenyhvuzliv-interpoljacji.html>

17. Орлик С. Программная инженерия. Проектирование программного обеспечения. Software Design. - М., 2005. 12 с

18. [Электронный ресурс]. – Доступ: <https://www.jetbrains.com/ru-ru/webstorm/WebStorm>

19. Бибо Бер , Кац Йегуда jQuery. Подробное руководство по продвинутому JavaScript; Символ-плюс - М., 2017. - 624 с.

20. [Электронный ресурс]. – Доступ: <https://reactjs.org/> ReactJs

21. [Электронный ресурс]. – Доступ: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н.. Азаров О.Д.

" " 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

“Методи аналізу даних в комп'ютерних системах”

08-23.МКР.006.00.003.ТЗ

Науковий керівник: доцент к.т.н.

_____ Черняк О. І.

Студент групи 2КІ-20м

_____ Проценко В.О.

ВНТУ 2022

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Важливим є актуальність дослідження у напрямку магістерської роботи, яка обумовлена тим, що одним з прикладних аспектів застосування математичних моделей є інформаційні системи, які збирають та обробляють великі масиви даних, та використовуються для підтримки вибору певних управлінських рішень.

При цьому коло завдань, що потребують вирішення, безперервно розширюється. Поряд з такими завданнями, як аналіз поведінки людини в веб-середовищі, дозволяє комерційним компаніям коригувати свої веб-сайти та бізнес-цілі, виявляючи області для вдосконалення, знаходження нових інструментів онлайн-просування сайту, тестування нових функціональних можливостей та, зрештою, збільшення власних прибутків.

1.2 Наказ про затвердження теми МКР.

2 Мета МКР і призначення розробки

2.1 Мета роботи — розробка алгоритмічного та програмного забезпечення апроксимації даних вимірювань аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з мережі Інтернет за допомогою попереднього використання алгоритмів інтерполяції;

2.2 Призначення розробки — відображення середнього часу перебування користувачів на сайті у визначенні проміжки часу, на основі даних про час та дату входу на сайт, а також час та дату виходу з сайту, усіх відвідувачів за певний період часу.

3 Вихідні дані для виконання МКР

3.1 Проведення збору та аналізу даних про використання веб-сайту;

3.2 Розробка імітаційної моделі веб-середовища та поведінки користувача;

3.2 Сегментування трафіка за активністю, залученістю і типу відвідувача;

3.3 Розробка веб-додатку на react js, у середовищі розробки Web Storm, що

відображає середній час перебування користувачів на сайті у визначенні проміжки часу;

3.4 Виконання розрахунків для доведення доцільності нової розробки з економічної точки зору.

4 Вимоги до виконання МКР

Головна вимога — використати, як основний, метод обробки даних — апроксимацію функцій однієї і двох змінних та інтерполяцію за формулою Лагранжа для створення програмного інструменту веб-аналітики.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Огляд сучасних методів даних в комп'ютерних системах	17.01.2022	31.01.2022	Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Збір даних з веб-середовища для розробки системи веб-аналітики	01.02.2022	14.02.2022	Розділ 2
3	Розробка програмного та технічного забезпечення	15.02.2022	31.03.2022	Розділ 3
4	Дослідження ефективності, проведення А/В тестування	01.04.2022	15.04.2022	Розділ 4
5	Підготовка економічної частини	16.04.2022	25.04.2022	Розділ 5
6	Апробація та впровадження результатів дослідження	09.05.2022	31.05.2022	Тези доповідей
	Опублікування результатів досліджень	03.06.2022	03.06.2022	Стаття
8	Оформлення пояснювальної записки, графічного матеріалу і презентації	03.06.2022	06.06.2022	Пояснювальна записка, графічний матеріал і презентація
8	Підготовка супроводжуючих документів, їх підписування, проходження нормоконтролю та тесту на плагіат	06.06.2022	06.06.2022	Оформлені документи

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки. Кафедра обчислювальної техніки 2022;

— Документами на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК Б

Лістинг програми (головний клас програми)

App.jsx

```
import React from 'react';
import './App.css';
import BarChart from "../components/BarChart";

export default function App() {

  return (
    <div className="main">
      <BarChart/>
    </div>
  );
}
```

ДОДАТОК В

Лістинг програми (Клас з логікою роботи графіка)

BarChart.jsx

```
import React, {useState} from 'react';
```

```
import {Line} from 'react-chartjs-2';
```

```
import data from '../Data.json';
```

```
import moment from "react-moment";
```

```
function BarChart(props) {
```

```
//PREPARE DEFINED VALUES
```

```
    const definedValues = new Map();
```

```
    const definedKeys = [];
```

```
    for(let i = 0; i < data.length; i++){
```

```
        let start = Date.parse(data[i].startDate + " " + data[i].startTime);
```

```
        let end = Date.parse(data[i].endDate + " " + data[i].endTime);
```

```
        let difference = end - start;
```

```
        let minutesDifference = Math.floor(difference/1000/60);
```

```
        definedKeys.push(start);
```

```
        definedValues.set(start, minutesDifference);
```

```
    }
```

```
//PREPARE DEFINED VALUES
```

```
//CALCULATE RANGE BETWEEN FIRST AND LAST ITEMS AND ADD TO LIST
```

```
let allStamps = [];  
  
let firstStamp = Date.parse(data[0].startDate + " " + "00:00")  
let lastStamp = Date.parse(data.slice(-1)[0].startDate + " " + "23:59")  
while (firstStamp < lastStamp){  
  allStamps.push(firstStamp);  
  firstStamp += 21600000;  
}
```

```
//CALCULATE RANGE BETWEEN FIRST AND LAST ITEMS AND ADD TO  
LIST
```

```
//LAGRANGE INTERPOLATION
```

```
const [dataMap, setDataMap] = useState([]);
```

```
class Data  
{  
  constructor(x,y)  
{  
  this.x=x;  
  this.y=y;  
}  
}
```

```
function interpolate(f,xi,n)
```

```

{
  let result = 0; // Initialize result

  for (let i = 0; i < n; i++)
  {
    // Compute individual terms of above formula
    let term = f[i].y;
    for (let j = 0; j < n; j++)
    {
      if (j !== i)
        term = term*(xi - f[j].x) / (f[i].x - f[j].x);
    }

    // Add current term to result
    result += term;
  }

  return result;
}

```

```

let f= []
for(let x = 0; x < definedKeys.length; x++)
{
  f.push(new Data(definedKeys[x], definedValues.get(definedKeys[x])))
}

```

```

console.log("Value of f(3) is : " +

```

```

        interpolate(f, allStamps[l], 2));

let allData = [];
for(let l = 0; l < allStamps.length; l++){
    allData.push(new Date(interpolate(f, allStamps[l],
f.length)).getUTCMinutes());
}

console.log(allData);
//LAGRANGE INTERPOLATION

//CONVERT TIMESTAMPS TO DATE FORMAT
let allDate = []

for(let t = 0; t < allStamps.length; t++){
    let date = new Date(+allStamps[t]);
    allDate.push(date.toDateString() + " " + date.toTimeString())
}

console.log(allDate)

//CONVERT TIMESTAMPS TO DATE FORMAT

return (
    <div>

```

```
<Line
  data={{
    labels: allDate,
    datasets: [{
      label: '# time in minutes',
      data: allData,
    }]
  }}

  height={600}
  width={400}
  options={{maintainAspectRatio: false}}

/>
</div>
);
}

export default BarChart;
```


ДОДАТОК Г

Лістинг програми (Файл стилів програми)

App.css

```
.App {
```

```
  text-align: center;
```

```
}
```

```
.App-logo {
```

```
  height: 40vmin;
```

```
  pointer-events: none;
```

```
}
```

```
@media (prefers-reduced-motion: no-preference) {
```

```
  .App-logo {
```

```
    animation: App-logo-spin infinite 20s linear;
```

```
  }
```

```
}
```

```
.App-header {
```

```
  background-color: #282c34;
```

```
  min-height: 100vh;
```

```
  display: flex;
```

```
  flex-direction: column;
```

```
  align-items: center;
```

```
  justify-content: center;
```

```
  font-size: calc(10px + 2vmin);
```

```
  color: white;
```

```
}
```

```
.main {  
  padding-left: 200px;  
  padding-right: 200px;  
}
```

```
.App-link {  
  color: #61dafb;  
}
```

```
@keyframes App-logo-spin {  
  from {  
    transform: rotate(0deg);  
  }  
  to {  
    transform: rotate(360deg);  
  }  
}
```

ДОДАТОК Д

Лістинг програми (Файл конфігурації програми)

```
package.json
{
  "name": "magistrjs",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.0.1",
    "@testing-library/user-event": "^13.5.0",
    "chart.js": "^2.9.4",
    "moment-timezone": "^0.5.34",
    "react": "^18.0.0",
    "react-chartjs-2": "^2.9.0",
    "react-dom": "^18.0.0",
    "react-moment": "^1.1.2",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4",
    "xlsx": "^0.18.5"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
```

```
"extends": [  
  "react-app",  
  "react-app/jest"  
]  
},  
"browserslist": {  
  "production": [  
    ">0.2%",  
    "not dead",  
    "not op_mini all"  
  ],  
  "development": [  
    "last 1 chrome version",  
    "last 1 firefox version",  
    "last 1 safari version"  
  ]  
}  
}
```

ДОДАТОК Е

Лістинг програми (Файл з даними про користувачів)

Data.json

```
[  
  {  
    "userId": "40",  
    "startDate": "2022-04-08",  
    "startTime": "01:01",  
    "endDate": "2022-04-08",  
    "endTime": "01:05"  
  },  
  {  
    "userId": "45",  
    "startDate": "2022-04-08",  
    "startTime": "02:11",  
    "endDate": "2022-04-08",  
    "endTime": "02:17"  
  },  
  {  
    "userId": "25",  
    "startDate": "2022-04-08",  
    "startTime": "02:36",  
    "endDate": "2022-04-08",  
    "endTime": "02:49"  
  },  
  {  
    "userId": "3",  
    "startDate": "2022-04-08",
```

```
"startTime": "03:48",
"endDate": "2022-04-08",
"endTime": "03:51"
},
{
  "userId": "4",
  "startDate": "2022-04-08",
  "startTime": "04:28",
  "endDate": "2022-04-08",
  "endTime": "05:01"
},
{
  "userId": "5",
  "startDate": "2022-04-09",
  "startTime": "01:12",
  "endDate": "2022-04-09",
  "endTime": "01:35"
},
{
  "userId": "5",
  "startDate": "2022-04-09",
  "startTime": "02:22",
  "endDate": "2022-04-09",
  "endTime": "02:35"
},
{
  "userId": "6",
```

```
"startDate": "2022-04-09",
"startTime": "03:15",
"endDate": "2022-04-09",
"endTime": "03:20"
},
{
  "userId": "14",
  "startDate": "2022-04-09",
  "startTime": "04:22",
  "endDate": "2022-04-09",
  "endTime": "04:27"
},
{
  "userId": "7",
  "startDate": "2022-04-09",
  "startTime": "04:38",
  "endDate": "2022-04-09",
  "endTime": "04:45"
},
{
  "userId": "22",
  "startDate": "2022-04-09",
  "startTime": "05:44",
  "endDate": "2022-04-09",
  "endTime": "05:53"
},
{
```

```
"userId": "23",
"startDate": "2022-04-09",
"startTime": "06:24",
"endDate": "2022-04-09",
"endTime": "06:35"
},
{
"userId": "13",
"startDate": "2022-04-10",
"startTime": "10:50",
"endDate": "2022-04-10",
"endTime": "10:54"
},
{
"userId": "12",
"startDate": "2022-04-10",
"startTime": "11:10",
"endDate": "2022-04-10",
"endTime": "11:17"
}
]
```


ДОДАТОК Ж

Лістинг програми (Головний html файл програми)

Index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

```
<meta name="theme-color" content="#000000" />
```

```
<meta
```

```
  name="description"
```

```
  content="Web site created using create-react-app"
```

```
/>
```

```
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
```

```
<!--
```

```
  manifest.json provides metadata used when your web app is installed on a
  user's mobile device or desktop. See
```

```
https://developers.google.com/web/fundamentals/web-app-manifest/
```

```
-->
```

```
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
```

```
<!--
```

```
  Notice the use of %PUBLIC_URL% in the tags above.
```

```
  It will be replaced with the URL of the `public` folder during the build.
```

```
  Only files inside the `public` folder can be referenced from the HTML.
```

```
  Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
  work correctly both with client-side routing and a non-root public URL.
```

Learn how to configure a non-root public URL by running ``npm run build``.

-->

```
<title>React App</title>
```

```
</head>
```

```
<body>
```

```
<noscript>You need to enable JavaScript to run this app.</noscript>
```

```
<div id="root"></div>
```

```
<!--
```

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the `<body>` tag.

To begin the development, run ``npm start`` or ``yarn start``.

To create a production bundle, use ``npm run build`` or ``yarn build``.

-->

```
</body>
```

```
</html>
```

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА
НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: _____

Тип роботи: _____ **магістерська кваліфікаційна робота** _____
(БДР, МКР)

Підрозділ _____ **кафедра обчислювальної техніки** _____
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність _____ **82,3** _____ Схожість _____ **17,7** _____

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ **Захарченко С.М.** _____
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) _____ (прізвище, ініціали)

Керівник роботи _____ (підпис) _____ (прізвище, ініціали)