Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» грудня 2023 р.

Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «**МЕТОД ОЦІНКИ ЕФЕКТИВНОСТІ ВЕБСАЙТІВ**»

Захищено на засіданні
Атестаційної комісії № 42
протокол № __ від __.12.2023 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ **СКОБ Ю. О.**

Виконала:

студентка 2 курсу, групи КУ– 61
за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології.
Галузь знань 15 – Автоматизація та приладобудування
ЧЕПАК Анастасія Альбертівна_____

Керівник: к.т.н., доцент кафедри теоретичної та прикладної системотехніки
СТРІЛЕЦЬ Вікторія Євгенівна_____

Рецензент: к.т.н., в.о. зав. кафедри теоретичної та прикладної інформатики
МЕНЯЙЛОВ Євген Сергійович_____

Харків – 2023

АНОТАЦІЯ

Пояснювальна записка до магістерської атестаційної роботи складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 84 сторінки, із яких 64 сторінок основної частини з 23 рисунками, 2 таблицями, 25 найменуваннями списку використаних джерел та чотирма додатками.

Метою дослідження є підвищення якості моделі оцінки ефективності вебсайтів за рахунок використання алгоритмів машинного навчання.

Об'єкт дослідження – процес оцінки ефективності та якості вебсайтів на основі впровадження алгоритмів машинного навчання.

Предмет дослідження – моделі та методи машинного навчання для оцінки ефективності вебсайтів.

Проблема, яка вирішується в кваліфікаційній роботі полягає в тому, щоб зібрати унікальні дані реальних вебсайтів та за допомогою методів машинного навчання побудувати регресійну модель оцінки ефективності роботи веб-сайтів.

Область застосування – програмна реалізація методів статистичного аналізу. Результати роботи можуть бути використані для аналізу і оцінки ефективності роботи вебсайтів, подальшої оптимізації, пошуку перспектив для подальшого розвитку веб-ресурсів та виявлення проблем, що негативно впливають на систему.

Ключові слова: вебсайт, оцінка ефективності, методи машинного навчання, Python.

ABSTRACT

The explanatory note to the master's attestation work consists of an introduction, three sections, conclusions, a list of used sources and four appendices. The total volume of the work is 84 pages, of which 64 pages are the main part with 23 figures, 2 tables, 25 titles of the list of used sources and four appendices.

The purpose of the study is to develop a model for evaluating the performance of web resources and its further evaluation for the possibility of improving performance indicators.

The object of research is the process of evaluating the effectiveness and quality of websites based on the implementation of machine learning algorithms.

The subject of the study is machine learning models and methods for evaluating the effectiveness of websites.

The problem that is solved in the qualification work is to collect unique data of real websites and using machine learning methods to build a regression model for evaluating the effectiveness of websites.

The field of application is software implementation of statistical analysis methods. The results of the work can be used to analyze and evaluate the effectiveness of websites, further optimization, search for prospects for the further development of web resources and identify problems that negatively affect the system.

Keywords: website, performance evaluation, machine learning methods, Python.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ…….....6
ВСТУП……..7
РОЗДІЛ 1. КРИТЕРІЇ ОЦІНКИ ВЕБСАЙТІВ…….....9
1.1 Функціональність вебсайтів.……..……..……..……..……..……9
1.2 Структура вебсайтів.……..……..……..……..……..……..…….....10
1.2.1 Основні види структур вебсайтів…….11
1.2.2 Елементи структури вебсайтів…….15
1.3 Вимоги до вебсайтів……17
1.3.1 Види вимог……..17
1.3.2 Рекомендації щодо вимог……...18
1.3.3 Базові вимоги до вебсайтів……18
1.4 Критерії оцінки ефективності вебсайтів…….20
Висновки за розділом 1……..……..……..……..……..……..…….....21
РОЗДІЛ 2. МЕТОДИ ОЦІНКИ ЕФФЕКТИВНОСТІ ВЕБСАЙТІВ…….……23
2.1 Методи для оцінки ефективності вебсайтів.…….……23
2.2 Методи машинного навчання...……28
2.2.1 Linear Regression…….28
2.2.2 Ridge Regression……..30
2.2.3 Support Vector Machine……...32
2.2.4 Gradient Boosting…….35
2.2.5 Decision Trees……..……37
2.2.6 Random Forest...……...40
2.2.7 k-nearest neighbor...……..43
Висновки за розділом 2……43
РОЗДІЛ 3. РОЗРОБКА МОДЕЛІ ОЦІНКИ ЕФЕКТИВНОСТІ ВЕБСАЙТІВ…..48
3.1 Набір вхідних даних моделі.…….....48
3.1.1 Формування та формалізація вхідних даних...……48

3.1.2 Попередня обробка даних……..49
3.1.3 Вибір параметрів вебсайтів для побудови моделі…….....50
3.2 Вибір програмного комплексу для побудови моделі……..51
3.3 Метрики оцінки моделей регресії…….52
3.3.1 Середня абсолютна похибка……..52
3.3.2 Середня квадратична помилка……...53
3.3.3 Середньоквадратична помилка……..54
3.3.4 R квадрат……..55
3.4 Навчання моделі…….56
3.4.1 Навчальна та тестова вибірки……56
3.4.2. Метод score……..56
3.4.3 Функція GreadSearchCV……..57
3.5 Результати дослідження…….58
3.5.1 Результати оцінки роботи моделей……58
3.5.2 Результати покращення ефективності моделі……...59
3.5.3 Результати оцінки роботи моделей……61
Висновки за розділом 3……...……...……...……...……65
ВИСНОВКИ…….....66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……..67
ДОДАТКИ……..70
ДОДАТОК А…….…..70
ДОДАТОК Б……....72
ДОДАТОК В……75
ДОДАТОК Г……81

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

UX – User Experience;

SSL – Secure Sockets Layer;

HTTPS – HyperText Transfer Protocol Secure;

SEO – Search Engine Optimization;

KNN – K-Nearest Neighbour;

SVM – Support Vector Machine;

HTML – HyperText Markup Language;

ML – Machine Learning;

AI – Artificial Intelligence;

MAE – Mean Absolute Error;

MSE – Mean Square Error;

RMSE - Root Mean Square Error.

ВСТУП

Інтернет-технології використовуються різними організаціями, компаніями та окремими людьми для ведення бізнесу, охорони здоров’я, банківської справи, створення і розповсюдження реклами, освітніх проектів та інших послуг. За останні кілька десятиліть було спроектовано, розроблено та впроваджено багато вебсайтів. Однак більшість з них не відповідають набору стандартів якості. Існує потреба в підході до оцінки ефективності та якості вебсайтів.

Ефективність більшої кількості вебсайтів є невисока через слабку структурованість, незручність використання, а також неякісну функціональність. Тому виявлення таких проблем є **актуальним** завданням, вирішення якого лежить у сфері розробки засобів автоматизованої оцінки ефективності вебсайтів. Задача оцінки роботи інтернет-ресурсів є комплексною в силу різноманіття вебсайтів та мов їх створення. Разом з тим, показники ефективності мають ряд особливостей, більшість з яких пов’язана із слабким математичним обґрунтуванням та формальним представленням. Саме для вирішення цієї задачі пропонується застосувати метод оцінки ефективності вебсайтів із використанням алгоритмів машинного навчання.

Метою дослідження є розробка моделі для оцінки ефективності веб-ресурсів та її подальший аналіз щодо можливості покращення показників ефективності.

Об'єкт дослідження – процес оцінки ефективності та якості вебсайтів на основі впровадження алгоритмів машинного навчання.

Методи дослідження: методи статистичного аналізу, методи машинного навчання, методи кількісного та якісного аналізу, методи аналізу інформативності.

Предмет дослідження – моделі та методи машинного навчання для оцінки ефективності вебсайтів.

Завдання дослідження:

1. Постановка задачі для оцінки ефективності роботи вебсайтів.
2. Збір та попередня обробка вхідних даних, їх формалізація й подальший кількісний та якісний аналіз.
3. Аналіз методів для оцінки ефективності роботи вебсайтів.
4. Побудова моделей оцінки ефективності за допомогою методів машинного навчання та подальше покращення їх показників.
5. Аналіз отриманих результатів та висновки.

РОЗДІЛ 1
КРИТЕРІЇ ОЦІНКИ ВЕБСАЙТІВ

У сучасному цифровому світі зростаюча кількість користувачів всесвітньої мережі Інтернет зробила якість зв’язку та веб-серверів важливим фактором для доступу до онлайн-сервісів і збільшення клієнтської бази організацій. Досягнення інформаційних технологій та Інтернету відкрили нові виміри в галузі електронної комерції. Традиційний спосіб роботи офлайн бізнесу переведено на онлайн-платформу, а керування ефективним вебсайтом електронної комерції для залучення клієнтів та просування бізнесу є запорукою успіху для будь-якої галузі.

Тому дизайн вебсайту повинен бути простим у використанні. Також він повинен мати якісну функціональність, щоб бути корисним для більшості відвідувачів. Зрештою, сайт повинен мати певний баланс між зручністю використання та функціонуванням, щоб максимально задовольнити користувачів.

Відвідувачі повинні мати доступ до функцій, які допомагають їм виконувати завдання, але вони також повинні мати можливість легко знаходити ці функції. Крім того, вони повинні вміти розуміти, як ними користуватися.

1.1 Функціональність веб-сайтів

Функціональність визначається тим, яким чином вебсайт може допомогти досягти цілей користувачу.

Відповідно до загальноприйнятих принципів висока функціональність – це здатність відвідувача, з мінімальними здібностями, користуватися системою, не почуваючись розгубленим. Таким чином, функціональність сайту визначається інтуїтивністю, а також простотою навігації.

Користувачі, які заходять на сайт вперше, не знайомі з його структурою. Якщо на ньому немає простої, інтуїтивно зрозумілої навігації, вони швидко загубляться в пошуку та покинуть такий сайт. Ось чому тема функціональності вебсайтів є такою актуальною. З іншого боку, якщо вони за короткий час зорієнтуються на сайті, зрозуміють його логічне розташування та легко знайдуть потрібну інформацію – це означає, що сайт працює.

Основні принципи функціональності:

– підтримка чіткої ієрархії елементів на сайті – зв’язки між елементами повинні відчуватися інтуїтивно. Чим важливіший елемент, тим помітнішим він має бути, наприклад, жирним шрифтом, характерним кольором або розташуванням на сторінці. Елементи, логічно пов'язані один з одним, повинні мати схожий стиль тексту та розміщення в одній групі;

– поділ сайту на функціональні зони – це дозволить користувачеві швидко та якісно знаходити потрібну інформацію. Він зможе швидко вирішити, яка частина сторінки містить корисний для нього вміст, а яку він може пропустити;

– позначення клікабельних елементів – дивлячись на певний елемент (посилання, кнопку, піктограму), кожен користувач має відразу знати, чи він клікабельний;

– використання загальновідомих умовностей, такі як логотип, який перенаправляє глядача на домашню сторінку, якщо натиснути значок кошика для покупок, який переносить користувача до списку покупок разом із значком лупи, який відповідає за пошукову систему сайту. Це знайомі звичайні методи швидкої навігації навіть для новачків;

– обмеження елементів, що відволікають увагу – занадто багато характерних елементів (барвисті банери, анімація тощо) можуть перевантажити глядача та відволікти увагу.

1.2 Структура вебсайтів

У наш час те, як будь-який бренд або бізнес виглядає в Інтернеті важливіше, ніж будь-коли. Створення ефектної взаємодії з користувачем часто залежить від звернення уваги на невидимі деталі, такі як структура вебсайту. Структура вебсайту допомагає користувачам орієнтуватися на сайтах і знаходити інформацію, яку вони шукають.

Структура вебсайту визначає ієрархію, порядок і організацію сторінок. Вебсайт об’єднує всі веб-сторінки за допомогою навігаційної системи меню, внутрішніх посилань і вмісту.

1.2.1 Основні види структур вебсайтів

Існують чотири основні структури вебсайтів:

- ієрархічна модель;
- послідовна (вона ж лінійна) модель;
- матрична модель;
- модель бази даних.

Найпопулярнішим типом структури вебсайту є ***ієрархічна модель***, де домашня сторінка є відправною точкою, розгалужуючись на різні категорії та сторінки відповідно до важливості. Оскільки ця структура універсальна, вона підходить для різних типів вебсайтів – від вебсайтів персональних послуг до онлайн-портфоліо.

По-перше, вміст повинен бути упорядкований за важливістю. У більшості випадків це означає, що користувач спочатку отримує загальну інформацію, перш ніж відкривати додаткові деталі

На рис. 1.1 ідеальне представлення ієрархічної структури вебсайту в онлайн-портфоліо Стівена Поповича. Зрештою, це полегшує навігацію для користувачів різними рівнями інформації та дій на сайті — починаючи з більших категорій, таких як «Краса» та «Реклама» на головній сторінці, відвідувачі отримують більш детальний пошук конкретних проектів і брендів.

Рисунок 1.1 – Приклад ієрархічної структури вебсайту (онлайн-портфоліо Стівена Поповича)

Послідовна веб-структура веде користувачів крок за кроком до досягнення їх мети, будь то звуження категорій, керування процесом пошуку чи допомога в пошуку форми реєстрації. Ця базова структура, яка не потребує обслуговування, підходить для сайтів із мінімальним вмістом і сторінками, як-от вебсайт малого бізнесу або онлайн-портфоліо.

Послідовна структура вебсайту починається на домашній або цільовій сторінці, де перераховано кілька сторінок або категорій. Відвідувачі слідують за лінійним потоком, який веде їх через подорож батьківськими сторінками, зрештою потрапляючи на бажаний вміст.

Елі Грей розробила свій фітнес-сайт, який показаний на рис. 1.2. Відображаючи чотири типи навчальних пакетів на її домашній сторінці, відвідувачі можуть вибрати, який з них найкраще, і сторінка за сторінкою розпочати процес реєстрації. Даний вебсайт є прикладом послідовної структури.

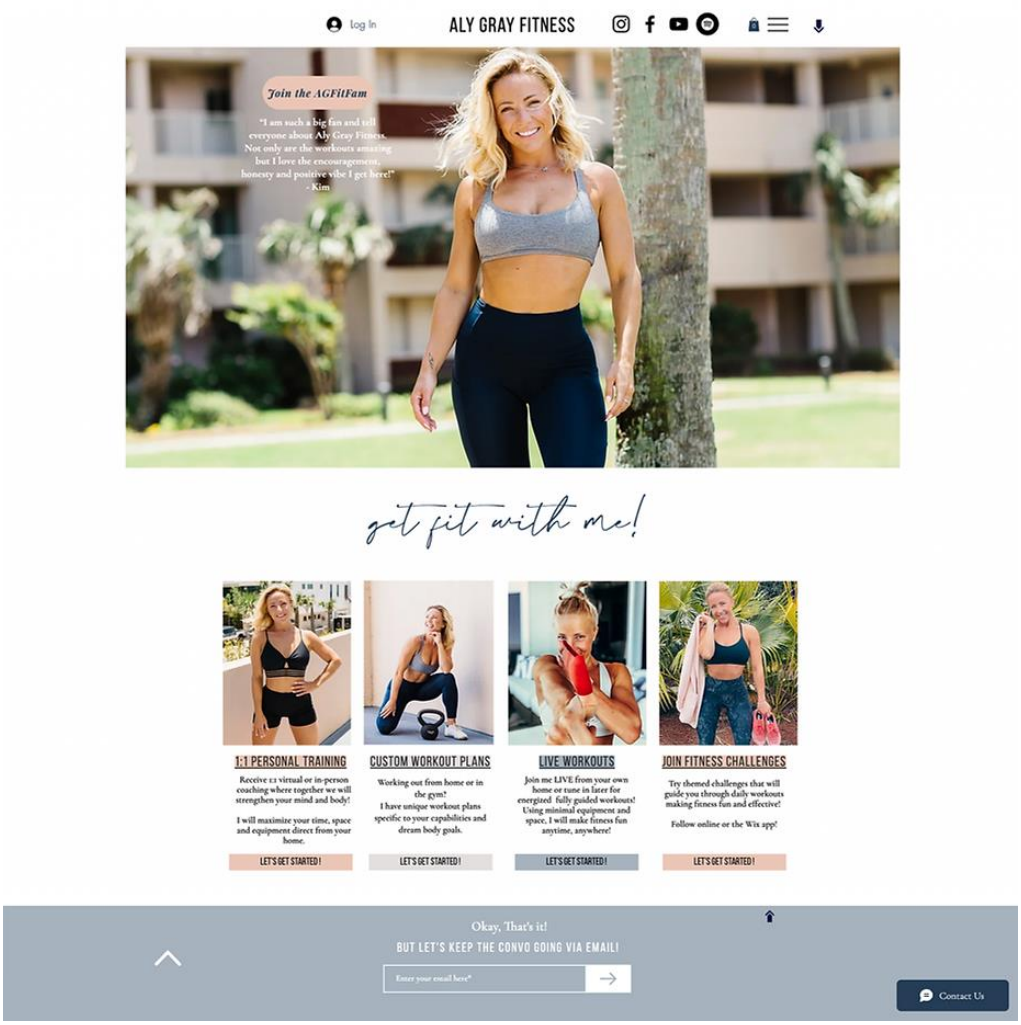

Рисунок 1.2 – Приклад послідовної структури вебсайту (фітнес-вебсайт Елі Грей)

Хоча ***матрична модель*** є однією з найстаріших моделей структури вебсайту, вона все ще залишається популярною сьогодні. Націлена на те, щоб дозволити відвідувачам насолоджуватися переглядом без суворих категорій. Це може здатися хаотичним, але для відвідувачів це означає повну свободу та багато точок входу до всього вмісту вебсайту.

Хоча категорії, підкатегорії та окремі сторінки все ще повинні існувати в матричній моделі, але не потрібно вибирати порядок їх появи для користувача. Натомість надійне взаємозв’язок вебсайту та підвищена важливість навігаційних функцій, таких як меню вебсайту чи рядок пошуку, сприяють взаємодії з користувачем.

Одними з найкращих прикладів матричних структур вебсайтів є онлайн-газети, онлайн-ресурси або великі вебсайти електронної комерції з різноманітним вмістом, пов’язаним один з одним. На рис. 1.3 показане використання матричної моделі на вебсайті Tach Clothing, використовуючи додавання навігації (і навігації по навігації) і спадне меню запропонованих «трендових продуктів» у рядку пошуку, щоб вивести відвідувачів за межі їхньої поточної подорожі — хоча й до відповідних пропозицій.

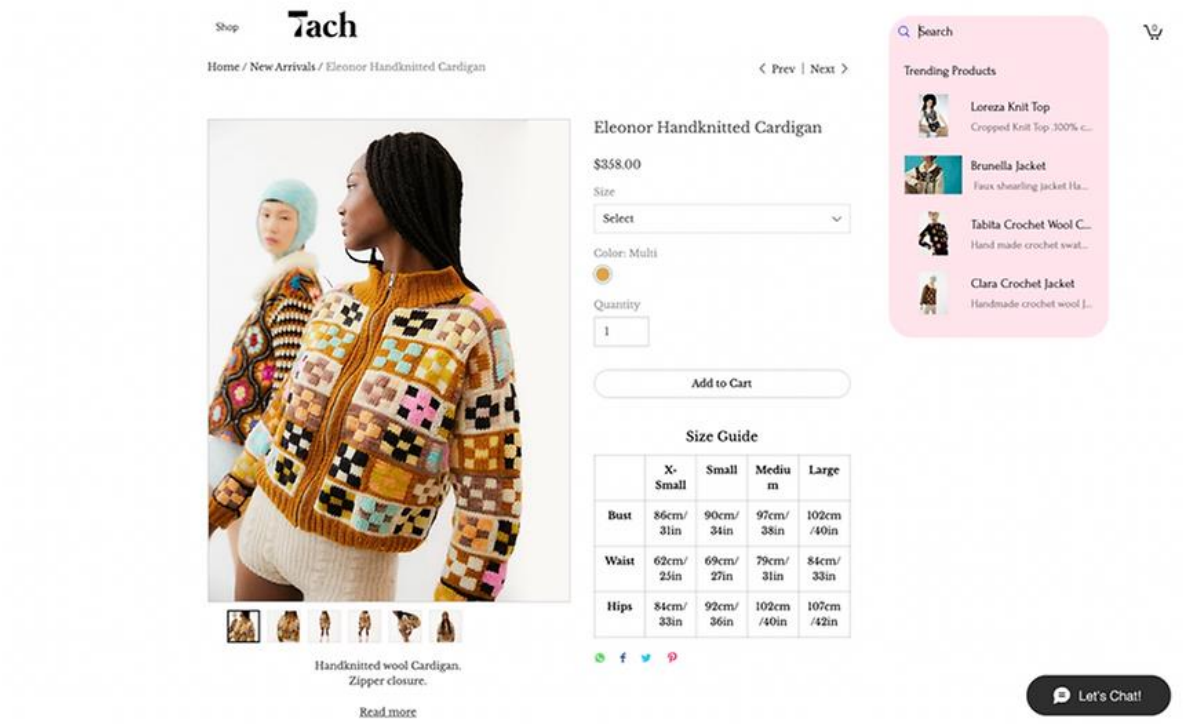

Рисунок 1.3 – Приклад матричної стурктури вебсайту (Tach clothing)

Структури вебсайтів, які відповідають ***моделі бази даних***, часто створюють динамічний і персоналізований досвід. Відвідувач сайту, який дотримується цієї моделі, як правило, повинен буде ввести свої особисті дані, запити або вподобання. Звідти вебсайт надасть їм відповідний вміст, який зберігається в базі даних веб-сторінки, як-от особисті дані або сторінки продуктів.

Бізнес-вебсайт для Code Zero Yachts (рис. 1.4) інтегрував складну систему даних, щоб допомогти відвідувачам знайти яхти, доступні в певний час, ціновий діапазон і місце розташування.

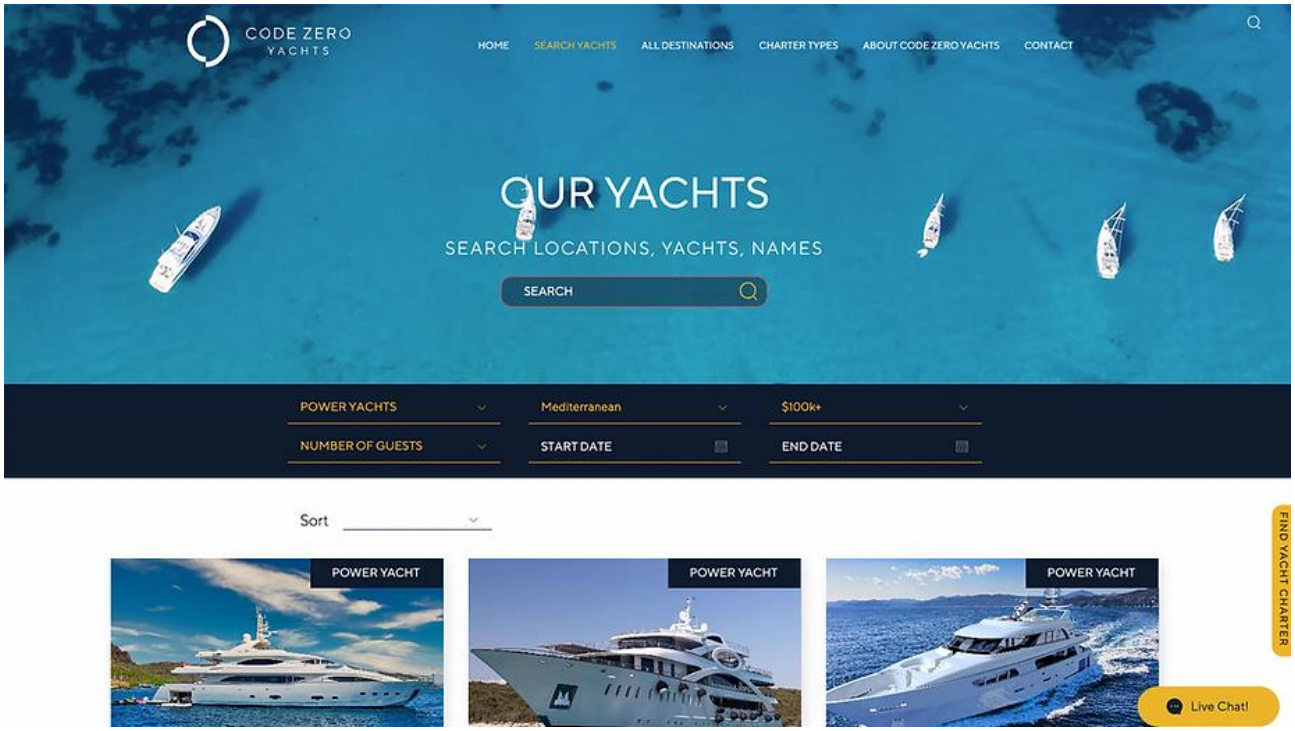

Рисунок 1.4 – Приклад використання моделі бази даних

1.2.2 Елементи структури вебсайтів

Разом із інформаційною архітектурою структура вебсайту є важливим фактором дизайну UX (користувацький досвід), який впливає на шлях відвідувача.

Визначення структури вебсайту гарантує, що відвідувачі слідуватимуть логічним шляхом користувача під час відкриття інформації вебсайту. Структура повинна включати всі веб-сторінки сайту, систему категорій для їх організації та засоби для відвідувачів, щоб переходити від одного елемента до іншого. Також структура сайту має гарантувати, що найважливішу інформацію можна знайти першою, водночас запрошуючи відвідувачів продовжувати досліджувати такі елементи:

- категорії та підкатегорії;

- навігація;
- система зв'язків.

Категорії створюють основу для організації структури вебсайту, групування сторінок сайту зі схожим вмістом і полегшення для відвідувачів пошуку того, що їм потрібно. Крім того, більшість сайтів, які мають велику кількість категорій, повинні додатково розділити свій вміст на підкатегорії.

Якщо сайт продає одяг, домашня сторінка вебсайту електронної комерції буде починатися з посилань на основні категорії покупок, наприклад, «взуття», «верхній одяг», «штани» або «аксесуари». Згодом ці сторінки спрямовуватимуть відвідувачів до різних підкатегорій, наприклад, «босоніжки» включатиметься в «взуття», а підкатегорія «тренч» — у «верхній одяг».

Навігація вебсайту прокладає його структуру для відвідувачів, діючи як карта вказівок до потрібного вмісту. У більшості випадків це починається з меню вебсайту, яке може бути класичним меню заголовка вгорі або більш мінімалістичним меню гамбургера.

Оскільки метою навігації є орієнтування відвідувачів, домашня сторінка має чітко відображати сторінки та категорії, які вони шукають на сайті. Звідти також мають бути доступні підкатегорії за допомогою спадного меню чи іншого способу відображення посилань на сторінки підкатегорій. Крім того, треба переконатися, що навігація сайту спрямовує відвідувачів на інші важливі веб-сторінки, пов’язані з вашим брендом, такі як сторінка «Про нас» або сторінка «Контакти».

Впровадження добре продуманої ***системи посилань*** (зв’язків) гарантує, що користувач переміщується сайтом належним чином. Залежно від типу структури вебсайту, яка використовується під час створення вебсайту, відвідувачі будуть залежати від системи посилань у різному ступені. Окрім посилань із меню вебсайту, структура може використовувати наведені нижче типи посилань для покращення взаємодії з користувачем:

– **з**аклики до дії – це стратегічно розміщені посилання, які спрямовують відвідувачів до прямої мети, заохочуючи їх зробити крок. Незалежно від того, чи

йдеться про посилання «зареєструватися» чи «придбати», заклики до дії надзвичайно корисні для відвідувачів, які мають на вашому сайті саме цю мету. Відображаються вони за допомогою жирного тексту або привабливого дизайну кнопки;

– внутрішні посилання – це посилання на вебсайті, які з'єднуються з іншими сторінками того самого сайту. Кожен сайт, природно, має внутрішні посилання між своїми веб-сторінками, чи організовано це найбільш оптимальним чином для відвідувачів, залежить виключно від розробників. Іноді компанії групують свої сторінки в категорії «кластери», використовуючи це як орієнтир. Наявність системи внутрішніх посилань також є найкращою практикою;

– контекстні посилання спрямовують відвідувача на пов'язаний вміст за межами веб-сторінок сайту, як-от сторінки продуктів інших компаній, публікації в блогах, джерела чи новини.

1.3 Вимоги до веб-сайтів

Вимоги до вебсайту – це список необхідних функцій, можливостей або характеристик, пов'язаних з вебсайтом і планами щодо його створення. Є кілька типів вимог, які можуть бути визначені під час процесу, які об'єднуються, щоб зосередити та визначити пріоритети плану проекту.

1.3.1 Види вимог

Існує багато різних типів документації вимог. На вищому рівні більшість може належати до однієї з категорій:

– бізнес-вимоги визначають цілі та проблеми, які зацікавлена сторона має намір вирішити за допомогою продукту;

– вимоги до користувача описують, як очікування користувача та як вони взаємодіятимуть із продуктом. Використовуються функції та вміст, описані у

сценаріях, для розробки вимог. Сценарії користувача мають окреслювати завдання, які користувачі хочуть виконати на сайті;

– функціональні вимоги надають деталі того, як повинен поводитися продукт, і визначають, що потрібно для розробки;

– вимоги до якості обслуговування детально описують, які характеристики має підтримувати продукт, щоб підтримувати свою ефективність і будь-які обмеження;

– вимоги до впровадження використовуються для деталізації змін у процесі, ролей команди, міграції з однієї системи в іншу тощо.

Вимоги до веб-сайту вказують лише те, що повинен мати вебсайт і що він повинен дозволяти користувачам робити. Вимоги не вказують, як спроектувати або розробити сайт, щоб мати ці функції та вміст.

1.3.2 Рекомендації щодо вимог

Вимоги можуть починатися як фраза або одне речення з описом того, що сайт повинен мати чи дозволяти користувачам, але вони стануть деталізовані у міру проходження процесу. Збір вимог може бути складним, але він допомагає забезпечити успіх проекту. Вони повинні бути:

– конкретні;

– цілком і добре продуманими;

– узгоджені з цілями, викладеними в керівних документах і статутах, і визначено пріоритети на основі них;

– такими, які можна перевірити під час тестування.

Вебсайт використовується як форма реклами, і він дозволяє потенційним клієнтам дізнатися більше про компанію та доступні продукти, які можуть бути запропоновані, однак є певні речі, які потрібні вебсайту, щоб переконатися, що він приносить результати.

1.3.3 Базові вимоги до вебсайтів

Основними вимогами до вебсайтів можна назвати такі:

– доменне ім'я буде першим, що знадобиться при створенні вебсайту компанії. Це цифрова адреса, яку люди використовуватимуть для підключення до вебсайту;

– хостинг. Веб-хостинг — це хост/сервер, на якому розміщено вміст вебсайту в Інтернеті. Тип хостингу, який вибирається, залежить від кількості відвідувачів, які відвідують вебсайт, і кількості місця, яке знадобиться.

Найпоширеніші типи хостингу:

1) спільний хостинг: спільний сервер з іншими вебсайтами;

2) виділений хостинг: особистий сервер. Це часто використовується на дуже великих вебсайтах корпоративного рівня.

– швидкість вебсайту. Важливо, щоб сайт завантажувався швидко. Це одна з найважливіших вимог, коли мова йде про дизайн і розробку веб-сайту. Щоб забезпечити завантаження вебсайту протягом 15 секунд, треба уникати надмірного використання зображень, анімації, відео, флеш-дизайну та аудіо;

– безпека SSL. Google оголосив, що перехід на HTTPS — додавання 2048-бітного сертифіката ключа SSL на сайт дасть незначне підвищення в рейтингу. Google повідомляє, що це дає вебсайтам невелику перевагу в рейтингу, оскільки вважається лише «дуже легким сигналом» у загальному алгоритмі рейтингу. На основі своїх тестів Google стверджує, що це впливає на «менше 1% глобальних запитів», але додає, що вони «можуть вирішити посилити» сигнал, оскільки хочуть «заохотити всіх власників вебсайтів переходити з HTTP на HTTPS, щоб утримати всіх безпечно в Інтернеті;

– SEO. Важливим аспектом SEO є полегшення розуміння вебсайту як користувачами, так і роботами пошукових систем. Хоча пошукові системи стають дедалі складнішими, вони все ще не можуть бачити та розуміти вебсторінку так само, як людина. SEO допомагає системам визначити, про що йдеться на кожній сторінці, і як вона може бути корисною для користувачів.

– відстеження та аналітика. Інструмент відстеження аналітики, який потрібен, щоб зрозуміти, як відвідувачі взаємодіють із вебсайтом. Знання того,

як працює вебсайт і як відвідувачі використовують його, дасть набагато краще уявлення про те, де можна покращити його;

– кросбраузерність. Вебсайт має бути доступним для перегляду в усіх типах сучасних браузерів, таких як Chrome, Firefox, Mozilla та Safari;

– інтеграція соціальних мереж. Зараз соціальна мережа є такою важливою частиною повсякденного життя, тому важливо мати кнопки спільного доступу до соціальних мереж, щоб відвідувачам було легко ділитися вмістом.

1.4 Критерії оцінки ефективності вебсайтів

Дуже важливим в процесі роботи вебсайтів, особливо комерційних, є оцінка їх ефективності. Під оцінкою ефективності розуміють правило, за яким обрані показники ефективності порівнюють між собою або з деякою нормою, якщо вона існує або її можна встановити. Серед критеріїв оцінки ефективності вебсайтів є часто вживаними такі типи критеріїв:

- технічні критерії швидкодії, оптимізації мережевого трафіку та вимог до апаратних ресурсів;
- технічні критерії надійності та безпеки й ефективності технічної підтримки;
- естетичні та художні критерії;
- психологічні критерії;
- системні критерії глобальних середовищ (відвідуваність сайту, рейтинг сайту та ін.).

Опис груп критеріїв наведений у таблиці 1.1.

Таблиця 1.1

Опис груп критеріїв оцінки вебсайтів

Фаза	Група критеріїв	Ця група критеріїв оцінює/вимірює
Інтерфейс	Принципи графічного дизайну	Ефективне використання кольору, тексту, фону та іншої загальної графіки
	Графіка та мультимедіа	Ефективність графіки та мультимедіа, що використовуються на сайті
	Стиль і текст	Чи є текст стислим і релевантним, а стиль гарним.
	Гнучкість і сумісність	Ступінь, до якого інтерфейс розроблений для обробки винятків, наприклад, лише текстові версії сторінок
Навігація	Логічна структура	Організація та система меню сайту
	Простота використання	Легкість навігації для пошуку сторінок, які шукає користувач
	Пошукова система	Здатність пошукової системи легко знаходити правильні сторінки та надавати чіткі описи результатів пошуку
	Навігаційні потреби	Інші важливі аспекти навігації, такі як відсутність непрацюючих посилань і сторінки «у розробці».
Зміст	Інформація про продукт/послуги	Незалежно від того, чи точно та ретельно описані продукти/послуги
	Компанія та контактна інформація	Незалежно від того, чи легко знайти інформацію про компанію, її співробітників і директорів
	Якість інформації	Релевантність контенту на сайті
Надійність	Збережений профіль клієнта	Процес реєстрації та те, як компанія використовує збережений профіль клієнта
	Процес замовлення	Ефективність і простота використання процесу онлайн-замовлення
	Отримання замовлення	Дії компанії від розміщення замовлення до його доставки
	Обслуговування клієнтів	Як компанія спілкується та допомагає своїм онлайн-клієнтам
Технічність	Швидкість	Різні аспекти швидкості завантаження сайту
	Безпека	Системи безпеки та способи, які використовує компанія для захисту конфіденційності клієнтів на сайті
	Програмне забезпечення та база даних	Гнучкість щодо використання різного програмного забезпечення. Також переглядає програмне забезпечення для даних і системи передачі даних, що використовуються на сайті
	Проектування системи	Правильне функціонування сайту та його інтеграція з внутрішніми та зовнішніми системами

Висновки за розділом 1

Після огляду, аналізу вебсайтів та їх особливостей ми можемо сказати, що вебсайт – це сукупність загальнодоступних взаємопов’язаних веб-сторінок, які мають одне доменне ім’я. Вебсайти можуть створюватися та підтримуватися окремою особою, групою, компанією чи організацією для різноманітних цілей. Веб-сторінки веб-сайту пов’язані за допомогою гіперпосилань і гіпертексту та мають спільний інтерфейс і дизайн. Вебсайт також може містити деякі додаткові документи та файли, такі як зображення, відео та інше.

З Інтернетом, що впливає на усі сфери діяльності, бачимо вебсайти з різними цілями. Отже, також можна сказати, що вебсайт також можна розглядати як цифрове середовище, здатне надавати інформацію та рішення та сприяти взаємодії між людьми, місцями та речами для підтримки цілей організації, для якої він був створений.

Існує ряд вимог до вебсайтів. Вимоги до веб-сайту вказують на те, що повинен мати вебсайт і що він повинен дозволяти користувачам робити. На роботу веб-ресурсів особливо впливає їх функціональність та структура. Тому дуже важливим в процесі роботи вебсайтів, особливо комерційних, є оцінка їх ефективності. Під оцінкою ефективності розуміють правило, за яким обрані показники ефективності порівнюють між собою або з деякою нормою.

РОЗДІЛ 2
МЕТОДИ ОЦІНКИ ЕФФЕКТИВНОСТІ ВЕБСАЙТІВ

2.1 Методи для оцінки ефективності вебсайтів

Бажаними якостями вебсайту є висока продуктивність, зручність використання, зручність для мобільних пристроїв, доступність, SEO (оптимізація для пошукових систем), зв'язок із соціальними мережами та захист. Щоб отримати інформацію, необхідну для покращення рейтингу вебсайту, запропонованого пошуковою системою, оцінка ефективності є важливою. Відповідно, у цьому прикладі оцінюється продуктивність вебсайту косметики на основі трав за допомогою автоматизованих інструментів оцінки, наприклад SEOptimer, Website Grader і Qualidator.

Сьогодні в Інтернеті з'явилися різноманітні джерела даних, і кожен день додається багато сайтів. Пошукові системи стали необхідним інструментом для користувачів, коли вони хочуть шукати інформацію в Інтернеті. Тим не менш, якщо вебсайт не має якості, необхідної для того, щоб ефективно відображатися у верхніх рейтингах сторінки в пошуковій системі в результаті запиту, це втрачена можливість, коли мова йде про доступ користувача до вебсайту [1]. Пошукова оптимізація – це спосіб, який справді реалізується для підвищення релевантності вебсайту з точки зору пошуку в основних пошукових системах [2]. Цей метод є хорошою практикою, коли йдеться про збільшення показу вебсайту, допомагаючи власникам досягти підвищеного рейтингу. Оптимізація продуктивності вебсайту та підвищення його зручності для мобільних пристроїв [3] має вирішальне значення для збільшення трафіку та покращення коефіцієнтів конверсії, а також створення нових можливостей для бізнесу таким чином, щоб швидко збільшити дохід. SEO корисно для покращення сайту [4] і передбачає редагування вебсайту з метою покращення його зовнішнього вигляду, наприклад, з точки зору структури вмісту та дизайну. Оцінка продуктивності

вебсайту є життєво важливою для стандарту вебсайту [5]. Відповідно, розгляд будь-яких проблем із юзабіліті, які існують щодо вебсайту, є проблемою, яку можна швидко вирішити. Інструменти зручності використання широко використовуються для перевірки якості цих вебсайтів, щоб покращити взаємодію з користувачами. Оцінка юзабіліті веб-сайту включає веб-тестування, щоб покращити його зручність для користувача та продуктивність, його зручність для мобільних пристроїв, його безпеку тощо. Користувачі отримують задоволення від використання доступних сайтів, які є інтуїтивно зрозумілими та простими для навігації, і припинять використовувати заплутані сайти [6, 7]. З усіх цих причин у цьому прикладі оцінюється ефективність конкретного вебсайту – cosmeticsotop.com – за допомогою автоматизованих інструментів оцінювання веб-сайту. Мета полягає в тому, щоб покращити продуктивність вебсайту щодо семи факторів: продуктивність, зручність використання, доступність, соціальні мережі, SEO, зручність для мобільних пристроїв і безпека.

Зазвичай автоматизовані інструменти оцінки вебсайту можуть перевіряти та звітувати про вебсайт щодо таких факторів, як продуктивність, зручність використання, зручність для мобільних пристроїв, доступність, оптимізація пошукової системи, соціальні мережі, безпека та рекомендації щодо редагування веб-сайту. Сім факторів для оцінки продуктивності вебсайту: [8]

- розмір сторінки;
- запити до сторінки;
- швидкість сторінки;
- кешування браузера;
- перенаправлення сторінок;
- стиснення.

Вісім факторів для оцінки вебсайту з точки зору пошукової оптимізації [9]:

- тег заголовка;
- мета-тег опису на сторінці;
- теги заголовка;

– теги заголовків HTML;

– узгодженість ключових слів;

– обсяг вмісту;

– атрибути Alt зображення;

– посилання має аспекти проблеми.

Два фактори для оцінки вебсайту з точки зору зручності для мобільних пристроїв:

– адаптивний дизайн призводить до підвищення рейтингу пошуку на мобільних пристроях;

– вікно перегляду керує сторінками з точки зору ширини та масштабу на різних типах пристроїв. Соціальні мережі для оцінки вебсайту: Facebook, Twitter, Instagram, YouTube, LinkedIn, Pinterest, Stumble on.

Безпека є центром оцінки вебсайтів у сертифікатах SSL для захисту вебсайтів від атак і впевненості в тому, що вебсайти справжні та надійні.

Оцінювання за допомогою інструмента SEOptimer:

SEOptimer — це засіб перевірки SEO вебсайту, який перевіряє продуктивність вебсайту, зручність використання, SEO, соціальні мережі та безпеку, щоб допомогти виявити проблеми, які можуть стримувати потенціал вебсайту. Крім того, він надає чіткий, дієвий, пріоритетний список рекомендацій, які допоможуть покращити якість вебсайту [9].

Оцінювання за допомогою Website grader Tool:

Інструмент оцінювання вебсайтів забезпечує загальну результативність вебсайту. Він перевіряє кожен аспект у формі розміру сторінки, запитів до сторінки, швидкості сторінки, кешування веб-переглядача.

Оцінка за допомогою інструменту Qualidator:

Інструмент Qualidator переглядає перші п’ять сторінок вебсайту, зазвичай використовуючи близько 60-70 автоматизованих тестів, відповідно, щодо основних аспектів зручності використання, доступності, SEO: пошукової оптимізації та технічної якості.

У прикладі, наведеному у табл. 2.1, розглядається вебсайт cosmeticsotop.com, який є вебсайтом рослинної косметики громадського підприємства в провінції Прачаупкіріхан у Таїланді, використані автоматизовані інструменти оцінки веб-сайту:

1) SEOptimer (www.seoptimer.com);
2) Website grader (www.website.grader.com);
3) Qualidator (www.qualidator.com).

Таблиця 2.1

Результат аналізу роботи сайту cosmeticsotop.com

	The Score of the Automated Evaluation Tools		
Criterion	SEOptimer	Website grader	Qualidator
1.Performance	56%	63.33%	65%
2.Usability & Mobile friendliness	100%	100%	70.5%
3.Accessibility	-	-	75.1%
4.SEO	57%	100%	67.1%
5.Social	50%	-	-
6.Security	71%	100%	-
overall	70.4%	89%	68.7%

На основі критеріїв аналізу вебсайту за 6 факторами [10, 11] продуктивність, зручність використання та зручність для мобільних пристроїв, пошукова оптимізація, соціальні мережі, безпека та загалом було оцінено вебсайт cometicsotop.com (табл. 2.1). Було використано три автоматичні інструменти оцінювання. При розгляді кожного інструменту інструмент SEOptimer отримав загальну оцінку 70,4%, під час розгляду кожного елемента було виявлено, що пункти найвищого рівня: «зручність використання та зручність для мобільних пристроїв» отримали оцінка 100%, «Безпека» отримала оцінку 71%, а пункт «SEO» отримав оцінку 57%. У випадку інструмента Website Grader із загальною оцінкою 89%, було виявлено, що елементи найвищого рівня були «Мобільний», «SEO» і «Безпека», кожен з яких отримав оцінку 100. Крім

того, у випадку інструмента Qualidator із загальною оцінкою 68,7 % було виявлено, що елемент найвищого рівня — «Доступність» має рейтинг 75,1 %, пункт «Юзабіліті» має рейтинг 70,5%, а «SEO» має рейтинг 67,1%.

Порівнюючи три інструменти для використання оцінки з точки зору SEO, соціальних мереж і безпеки, краще використовувати інструмент SEOptimer. Він має повний пункт для аудиту. Для оцінки продуктивності і зручності для мобільних пристроїв, краще використовувати Website grader Tool. Крім того, для оцінювання доступності та зручності використання, краще використати інструмент Qualidator.

Оцінювання за допомогою інструмента WebQEM:

Обсяг інформації, що поширюється в Інтернеті, зростає. Вебсайти та веб-додатки швидко розширилися. Вебсайти повинні відповідати високому стандарту, якщо вони часто використовуються для отримання інформації, необхідної для різних цілей. Кавіндра Кумар Сінгх та ін. [12] створили методологію, відому як WebQEM для обчислення якості вебсайтів на основі об’єктивної оцінки. Проте судження про якість вебсайтів на основі суб’єктивної оцінки може бути точнішим. Вони кількісно оцінили якість вебсайту на основі неупередженого огляду. Вони включили в свій метод якості, характеристики та підхарактеристики.

Процеси оцінювання та порівняння потребують як методологічної, так і технологічної підтримки.

WebQEM_Tool об’єднує елементи, щоб отримати схему та обчислює глобальний індикатор якості для кожного сайту. Це дозволяє оцінювачам оцінювати та порівнювати якість веб-продукту.

WebQEM_Tool покладається на веб-модель гіпердокументів, яка підтримує відстеження аспектів оцінювання. Він показує результати оцінювання за допомогою пов’язаних сторінок із текстовою, табличною та графічною інформацією та динамічно створює сторінки з цими результатами з таблиць, що зберігаються на рівні даних.

Процес WebQEM включає чотири основні технічні етапи:

1. Визначення та специфікація вимог до якості;
2. Елементарна оцінка (етапи проектування та впровадження);
3. Глобальна оцінка (етапи розробки та реалізації);
4. Висновок (рекомендації).

2.2 Методи машинного навчання

Машинне навчання або Machine Learning – різновид штучного інтелекту в основі якого лежить вивчення комп'ютерних алгоритмів, які автоматично навчаються в міру накопичення досвіду.

2.2.1 Linear Regression

Linear Regression (лінійна регресія) є різновидом контрольованого алгоритму машинного навчання, який навчається з позначених наборів даних і відображає точки даних на найбільш оптимізовані лінійні функції, які можна використовувати для прогнозування нових наборів даних.

Контрольовані алгоритми машинного навчання – це тип машинного навчання, де алгоритм вивчає дані з мітками. Дані з мітками означають набір даних, відповідне цільове значення якого вже відомо. Контрольоване навчання має два типи:

– класифікація: прогнозує клас набору даних на основі незалежної вхідної змінної. Клас - це категоричні або дискретні значення.

– регресія: передбачає безперервні вихідні змінні на основі незалежної вхідної змінної, наприклад, прогнозування цін на житло на основі різних параметрів, таких як вік будинку, відстань від головної дороги, розташування, площа тощо.

Лінійна регресія – це тип керованого алгоритму машинного навчання, який обчислює лінійний зв’язок між залежною змінною та однією чи кількома незалежними ознаками. Коли номер незалежної ознаки дорівнює 1, це називається однофакторною лінійною регресією, а у випадку більш ніж однієї

ознаки – багатовимірною лінійною регресією. Мета алгоритму — знайти найкраще лінійне рівняння, яке може передбачити значення залежної змінної (y) на основі незалежної змінної (x). Рівняння забезпечує пряму лінію, яка представляє зв’язок між залежною та незалежною змінними. Нахил лінії вказує на те, наскільки залежна змінна змінюється на зміну одиниці незалежної змінної (змінних).

Лінійна регресія використовується в багатьох різних областях, включаючи фінанси, економіку та психологію, щоб зрозуміти та передбачити поведінку певної змінної. Наприклад, у фінансах лінійна регресія може бути використана для розуміння зв’язку між ціною акцій компанії та її прибутком або для прогнозування майбутньої вартості валюти на основі її минулих показників.

Одним із найважливіших завдань навчання під наглядом є регресія. У регресії присутній набір записів зі значеннями X і Y, і ці значення використовуються для вивчення функції, тому, якщо передбачувати Y на основі невідомого X, можна використовувати цю вивчену функцію. У регресії шукають значення Y, тому потрібна функція, яка передбачає безперервний Y у випадку регресії з урахуванням X як незалежних ознак.

Тут Y називається залежною або цільовою змінною, а X — незалежною змінною, також відомою як предиктор Y. Існує багато типів функцій або модулів, які можна використовувати для регресії. Лінійна функція є найпростішим типом функції. Тут X може бути однією ознакою або декількома ознаками, що представляють проблему.

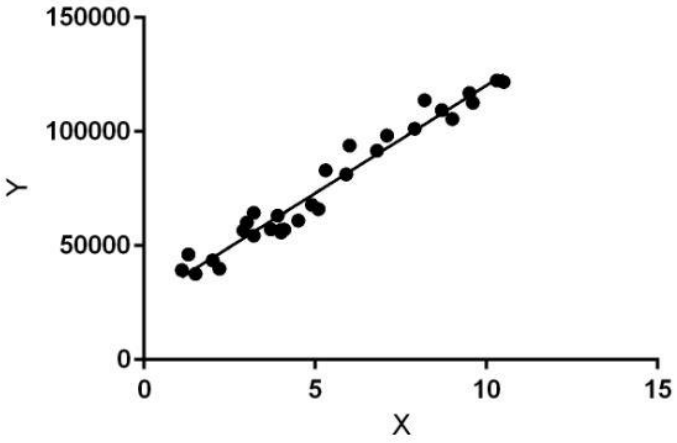

Рисунок 2.1 – Лінійна регресія

Лінійна регресія виконує завдання прогнозування значення залежної змінної (y) на основі заданої незалежної змінної (x)). Звідси й назва — лінійна регресія. На рисунку 2.1 X (вхід) — досвід роботи, а Y (вихід) — зарплата людини. Лінія регресії найкраще підходить для моделі.

Лінійна регресія є потужним інструментом для розуміння та прогнозування поведінки змінної, однак вона має відповідати декільком умовам, щоб бути точними та надійними рішеннями:

– лінійність: незалежна та залежна змінні мають лінійний зв’язок одна з одною. Це означає, що зміни в залежній змінній слідують за змінами в незалежній змінній (змінних) лінійним чином.

– незалежність: спостереження в наборі даних не залежать одне від одного. Це означає, що значення залежної змінної для одного спостереження не залежить від значення залежної змінної для іншого спостереження.

– нормальність: помилки в моделі розподілені нормально.

– відсутність мультиколінеарності: немає високої кореляції між незалежними змінними. Це вказує на те, що кореляція між незалежними змінними незначна або відсутня.

2.2.2 Ridge Regression

Ridge Regression (хребтова або гребенева регресія) – це спеціалізована техніка, яка використовується для аналізу даних множинної регресії, яка є мультиколінеарною за своєю природою.

Основна ідея хребтової регресії полягає в тому, що вона зосереджена на підгонці нової лінії.

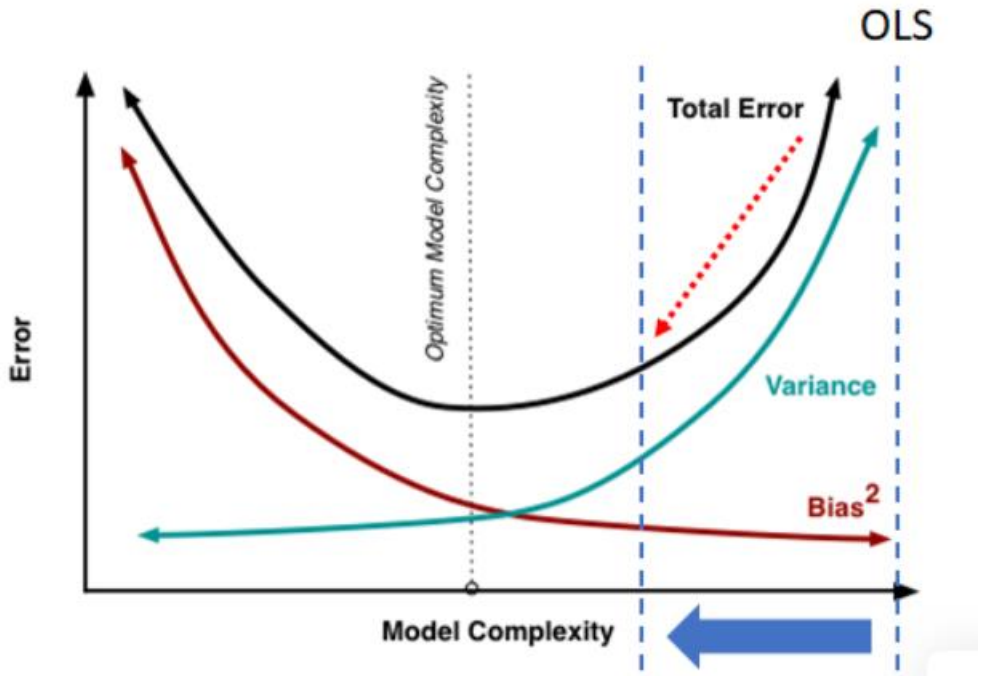

Рисунок 2.2 – Хребтова (гребенева) регресія

Хребтова регресія використовується з метою створення економних моделей, коли кількість змінних предиктора в даному наборі перевищує кількість спостережень або коли набір даних має мультиколінеарність. В основному використовується для аналізу мультиколінеарності в даних множинної регресії.

Мультиколінеарність – це явище, в якому одне прогнозоване значення в декількох регресійних моделей лінійно прогнозується з іншими, щоб досягти певного рівня точності. Мультиколінеарність по суті виникає, коли існує висока кореляція між більш ніж двома прогнозованими змінними. Вона означає існування кореляції між незалежними змінними в змодельованих даних. Це може

призвести до неточності в оцінках коефіцієнта регресії. Це може навіть збільшити стандартні помилки в коефіцієнтах регресії та знизити ефективність будь-яких тестів. Мультиколінеарність може спричинити оманливі результати та значення, що знижує ефективність і надійність її передбачуваності.

Якщо модель надмірно визначена, буде видно, що мультиколінеарність спричинена існуванням більшої кількості змінних, ніж спостережень. Це можливо уникнути цього під час розгортання моделі.

Переваги хребтової регресії:

- захищає модель від переобладнання;
- зміщення є достатнім, щоб зробити оцінки достатньо надійними наближеннями до справжніх значень;
- добре працює, коли є великі багатовимірні дані з кількістю предикторів більшою за кількість спостережень.
- оцінювач хребтової регресії дуже ефективний, коли мова йде про покращення оцінки за методом найменших квадратів у ситуаціях, коли існує мультиколінеарність;
- зменшує складність моделі.

Недоліки хребтової регресії:

- включає всі предиктори в остаточній моделі;
- не здатний виконувати вибір функцій;
- зменшує коефіцієнти до нуля;
- обмінює дисперсію на зміщення.

2.2.3 Support Vector Machine

Support Vector Machine (машина опорних векторів) також може бути стандартним інструментом машинного навчання для класифікації та регресії [13]. Машина опорних векторів може використовуватися як метод регресії, зберігаючи всі характеристики, що характеризують алгоритм (максимальний запас). Регресія опорного вектора використовує подібні основи для класифікації, що й SVM, лише з кількома незначними відмінностями, оскільки результатом є

фактичне число, тому стає дуже важко передбачити наявну інформацію, яка має безмежні можливості. Головна ідея: мінімізувати помилку, індивідуалізувати гіперплощину, яка максимізує маржу, маючи на увазі, що частина помилки допускається [14].

Алгоритм будує опорні вектори в області ознак великої розмірності. Потім будується гіперплощина з максимальним запасом. Функція ядра використовується для перетворення даних, що збільшує розмірність даних. Це збільшення стимулює те, що дані можуть бути розділені гіперплощиною з набагато вищою ймовірністю, і встановити мінімальну міру помилки ймовірності прогнозування [15]. На рисунку 2.3 показано основний процес машинного методу опорних векторів.

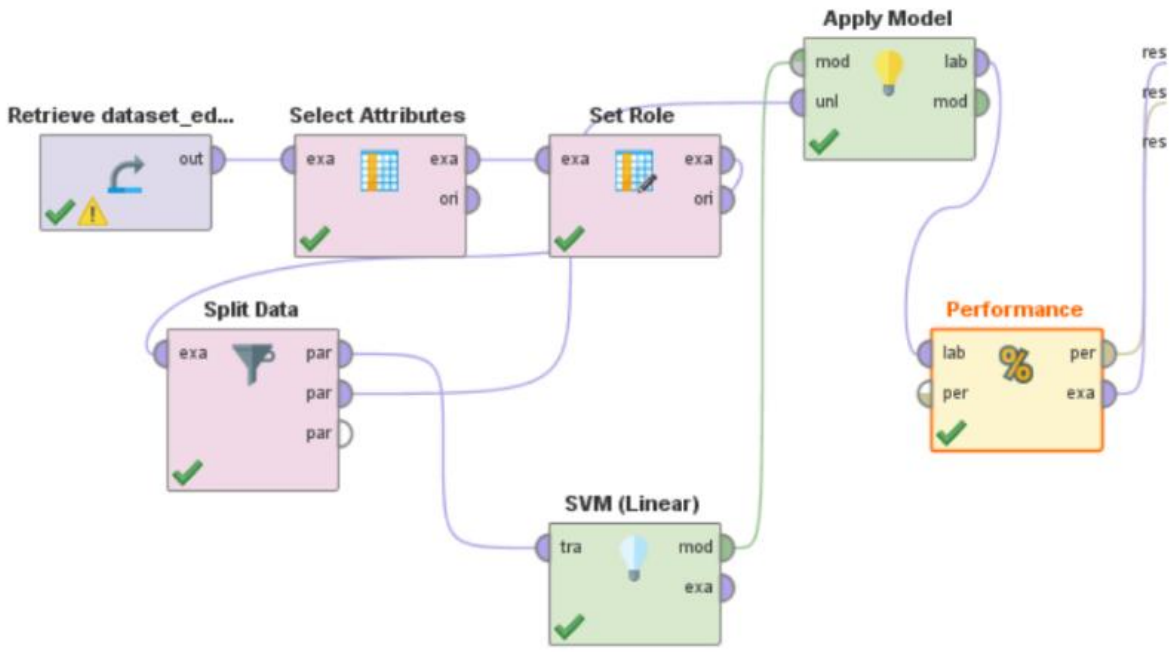

Рисунок 2.3 – Основний процес машинного методу опорних векторів

Support Vector Machine (SVM) — це потужний контрольований алгоритм машинного навчання з кількома перевагами. Деякі з основних переваг SVM включають:

- обробка даних великої розмірності: SVM ефективні в обробці даних великої розмірності, що є поширеним у багатьох програмах, таких як класифікація зображень і тексту;
- обробка невеликих наборів даних: SVM можуть добре працювати з невеликими наборами даних, оскільки їм потрібна лише невелика кількість опорних векторів для визначення межі;
- моделювання нелінійних меж прийняття рішень: SVM можуть моделювати нелінійні межі прийняття рішень за допомогою трюку ядра, який відображає дані у просторі з більшою розмірністю, де дані стають лінійно роздільними;
- стійкість до шуму: SVM стійкі до шуму в даних, оскільки межа рішення визначається опорними векторами, які є найближчими точками даних до межі;
- узагальнення: SVM мають хорошу продуктивність узагальнення, що означає, що вони здатні добре класифікувати нові, невидимі дані;
- універсальність: SVM можна використовувати як для завдань класифікації, так і для регресії, і його можна застосовувати для широкого спектру програм, таких як обробка природної мови, комп'ютерне бачення та біоінформатика;
- розріджене рішення: SVM мають розріджені рішення, що означає, що вони використовують лише підмножину навчальних даних для прогнозування. Це робить алгоритм більш ефективним і менш схильним до переобладнання;
- регуляризація: SVM можна регуляризувати, що означає, що алгоритм можна модифікувати, щоб уникнути переобладнання.

Support Vector Machine (SVM) — це потужний контрольований алгоритм машинного навчання, але він також має деякі обмеження та недоліки. Деякі з основних недоліків SVM включають:

- обчислювання дорого: SVM можуть бути обчислювально дорогими для великих наборів даних, оскільки алгоритм вимагає розв’язання задачі квадратичної оптимізації;
- вибір ядра: вибір ядра може значно вплинути на продуктивність SVM і може бути важко визначити найкраще ядро для певного набору даних;
- чутливість до вибору параметрів: SVM можуть бути чутливими до вибору параметрів, таких як параметр регулярізації і може бути важко визначити оптимальні значення параметрів для певного набору даних;
- витратність пам’яті: SVM можуть займати багато пам’яті, оскільки алгоритм потребує зберігання матриці ядра, яка може бути великою для великих наборів даних;
- обмежено проблемами двох класів: SVM переважно використовуються для проблем двох класів, хоча багатокласові проблеми можна вирішити за допомогою стратегій «один проти одного» або «один проти всіх»;
- відсутність імовірнісної інтерпретації: SVM не забезпечують імовірнісної інтерпретації межі рішення, що може бути недоліком у деяких програмах;
- не підходить для великих наборів даних із багатьма функціями: SVM можуть бути дуже повільними та споживати багато пам’яті, якщо набір даних має багато функцій;
- не підходить для наборів даних із відсутніми значеннями: SVM вимагає повних наборів даних без відсутніх значень, він не може обробляти відсутні значення.

2.2.4 Gradient Boosting

Gradient Boosting (градієнтне підсилення) — це популярний алгоритм підвищення в машинному навчанні, який використовується для завдань класифікації та регресії. Це один із методів ансамблевого навчання, який

послідовно навчає модель, і кожна нова модель намагається виправити попередню модель. Він поєднує кількох слабких учнів у сильних.

Градієнтне підсилення — це потужний алгоритм підвищення, який об’єднує кількох слабких учнів в одного сильного, у якому кожна нова модель навчена мінімізувати функцію втрат, таку як середня квадратична помилка або крос-ентропія попередньої моделі за допомогою градієнтного спуску. На кожній ітерації алгоритм обчислює градієнт функції втрат щодо прогнозів поточного ансамблю, а потім навчає нову слабку модель, щоб мінімізувати цей градієнт. Прогнози нової моделі потім додаються до ансамблю, і процес повторюється, доки не буде виконано критерій зупинки.

Підвищення градієнта оновлює вагові коефіцієнти, обчислюючи від’ємний градієнт функції втрат відносно прогнозованого результату.

Градієнтне підсилення може використовувати широкий спектр базових засобів навчання, таких як дерева рішень і лінійні моделі.

Підсилення градієнта зазвичай більш надійне, оскільки воно оновлює ваги на основі градієнтів, які менш чутливі до викидів.

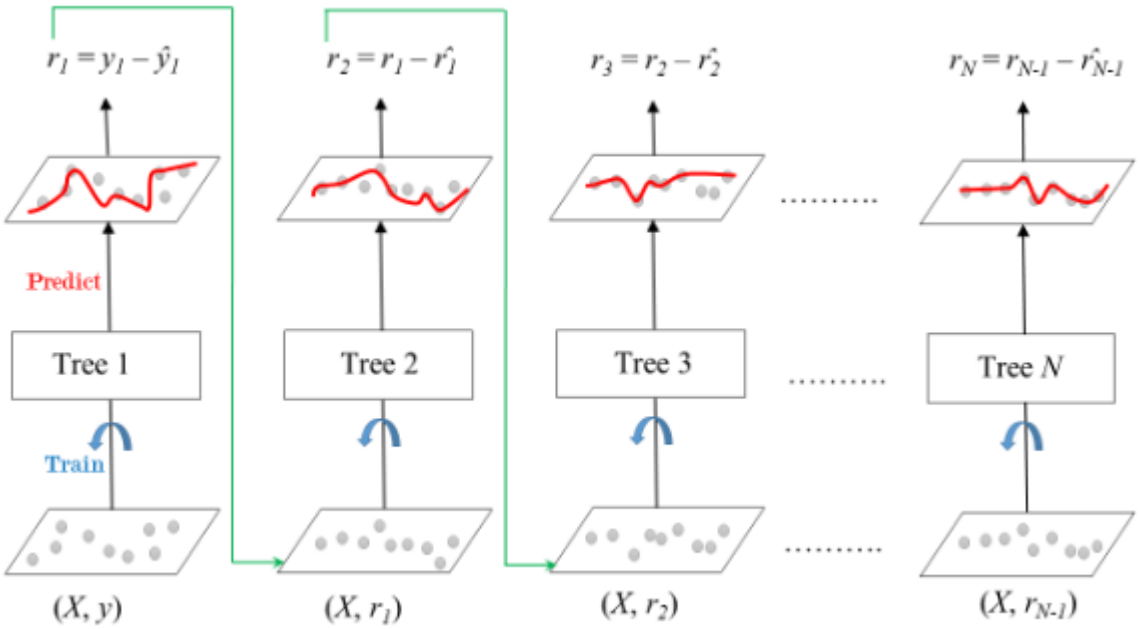

Рисунок 2.4 – Градієнтне підсилення

Переваги градієнтного підсилення:

– загалом більш точний порівняно з іншими режимами;

– тренуватися швидше, особливо на великих наборах даних;

– більшість із них забезпечують підтримку обробки категорійних функцій;

– деякі з них нативно обробляють відсутні значення.

Недоліки градієнтного підсилення:

– схильність до переобладнання: це можна вирішити шляхом застосування штрафів регуляризації L1 і L2;

– моделі можуть бути обчислювально дорогими і займати багато часу для навчання;

– важко інтерпретувати остаточні моделі.

2.2.5 Decision trees

Decision trees (дерева рішень) — це непараметричний контрольований алгоритм навчання, який використовується як для завдань класифікації, так і для

регресії. Він має ієрархічну структуру дерева, яка складається з кореневого вузла, гілок, внутрішніх вузлів і листових вузлів.

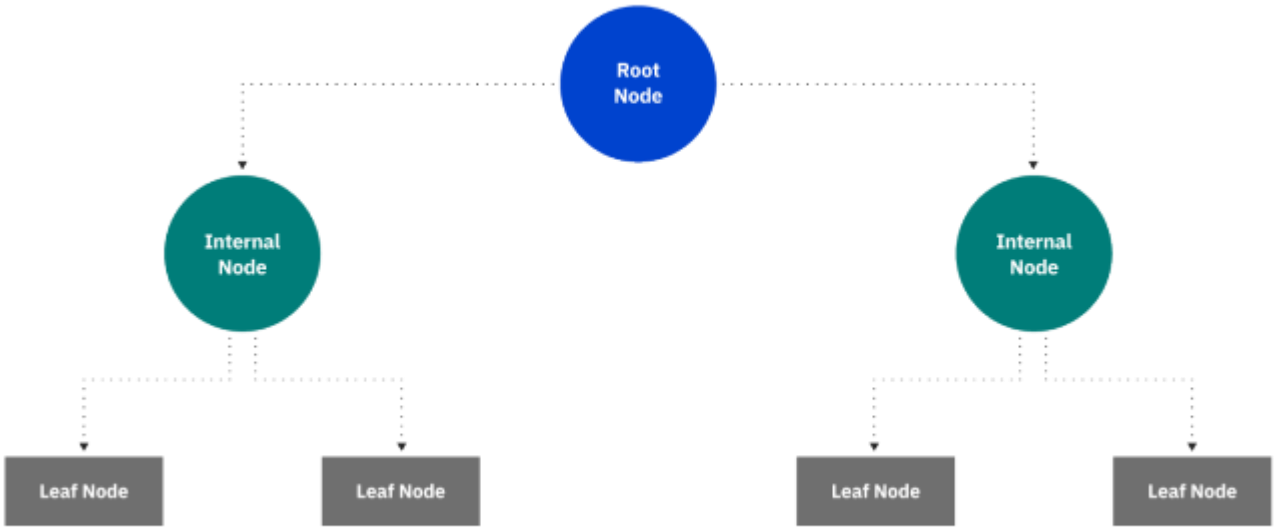

Рисунок 2.5 – Дерево рішень

Як показано на рис. 2.5, дерево рішень починається з кореневого вузла, який не має жодних вхідних гілок. Вихідні гілки від кореневого вузла потім надходять у внутрішні вузли, також відомі як вузли прийняття рішень. На основі доступних функцій обидва типи вузлів проводять оцінки для формування однорідних підмножин, які позначаються листовими вузлами або кінцевими вузлами. Листові вузли представляють усі можливі результати в наборі даних.

Навчання дерева рішень використовує стратегію «розділяй і володарюй», проводячи жадібний пошук, щоб визначити оптимальні точки поділу всередині дерева. Потім цей процес поділу повторюється рекурсивним способом зверху вниз, доки всі або більшість записів не буде класифіковано за певними мітками класу. Чи всі точки даних класифікуються як однорідні набори, значною мірою залежить від складності дерева рішень. Меншим деревам легше отримати чисті вузли листя, тобто точки даних в одному класі. Однак у міру збільшення розміру

дерева стає все важче підтримувати цю чистоту, і це зазвичай призводить до того, що в даному піддереві потрапляє занадто мало даних. Коли це відбувається, це називається фрагментацією даних, і це часто може призвести до переобладнання. Як наслідок, дерева рішень мають перевагу малим деревам, що узгоджується з принципом економії в Бритві Оккама; тобто «сутності не слід множити понад потребу». Інакше кажучи, дерева рішень повинні додавати складності лише за необхідності, оскільки найпростіше пояснення часто є найкращим. Щоб зменшити складність і запобігти переобладнанню, зазвичай використовується обрізка; це процес, який видаляє гілки, які розбиваються на функції з низьким значенням. Відповідність моделі потім можна оцінити за допомогою процесу перехресної перевірки.

Алгоритм Ханта, який був розроблений у 1960-х роках для моделювання навчання людини в психології, є основою багатьох популярних алгоритмів дерева рішень, таких як:

– ID3: Росс Куінлан бере участь у розробці ID3, що є скороченням від «Iterative Dichotomiser 3». Цей алгоритм використовує ентропію та приріст інформації як показники для оцінки розподілу кандидатів;

– C4.5: цей алгоритм вважається пізнішою ітерацією ID3, який також був розроблений Quinlan. Він може використовувати приріст інформації або коефіцієнти приросту для оцінки точок розділення в межах дерев рішень;

– CART: термін CART є абревіатурою від «дерева класифікації та регресії» і був введений Лео Брейманом. Цей алгоритм зазвичай використовує домішку Джіні для визначення ідеального атрибута для розділення. Домішка Джіні вимірює, як часто навмання вибраний атрибут неправильно класифікується. При оцінці за допомогою домішки Джіні більш ідеальним є нижче значення.

Переваги дерев рішень:

– легко інтерпретувати: булева логіка та візуальні представлення дерев рішень полегшують їх розуміння та використання. Ієрархічна природа дерева

рішень також дозволяє легко побачити, які атрибути є найважливішими, що не завжди зрозуміло з іншими алгоритмами, такими як нейронні мережі;

– підготовка даних майже не потрібна: дерева рішень мають низку характеристик, які роблять їх більш гнучкими, ніж інші класифікатори. Він може обробляти різні типи даних, тобто дискретні чи безперервні значення, а безперервні значення можна перетворити на категоричні значення за допомогою порогів. Крім того, він також може обробляти значення з відсутніми значеннями, що може бути проблематичним для інших класифікаторів, таких як Naive Bayes;

– більша гнучкість: дерева рішень можна використовувати як для завдань класифікації, так і для регресії, що робить його більш гнучким, ніж деякі інші алгоритми. Він також нечутливий до базових зв'язків між атрибутами; це означає, що якщо дві змінні сильно корельовані, алгоритм вибере лише одну з ознак для розділення.

Недоліки дерев рішень:

– схильність до переобладнання: складні дерева рішень, як правило, переобладнують і погано узагальнюють нові дані. Цього сценарію можна уникнути за допомогою процесів попередньо або після обрізки. Попереднє обрізання зупиняє ріст дерева, якщо даних недостатньо, тоді як постобрізання видаляє піддерева з невідповідними даними після створення дерева;

– оцінювачі високої дисперсії: невеликі варіації в межах даних можуть створити зовсім інше дерево рішень. Усереднення оцінок може бути методом зменшення дисперсії дерев рішень. Однак цей підхід є обмеженим, оскільки він може призвести до сильно корельованих предикторів;

– дороговартість: враховуючи, що дерева рішень під час побудови використовують жадібний підхід пошуку, навчання їх може бути дорожчим порівняно з іншими алгоритмами;

– не повністю підтримується в scikit-learn: scikit-learn — це популярна бібліотека машинного навчання на основі Python. Хоча ця бібліотека має модуль дерева рішень, поточна реалізація не підтримує категоріальні змінні.

2.2.6 Random forest

Алгоритми random forest (випадковий ліс) мають три основні гіперпараметри, які необхідно встановити перед навчанням. До них належать розмір вузла, кількість дерев і кількість відібраних функцій. Звідти класифікатор випадкового лісу можна використовувати для вирішення проблем регресії або класифікації.

Алгоритм випадкового лісу складається з набору дерев рішень, і кожне дерево в ансамблі складається із вибірки даних, отриманої з навчального набору із заміною, яка називається початковою вибіркою. З цієї навчальної вибірки одна третина відкладена як тестові дані, відомі як вихідна (oob) вибірка. Інший випадок потім вводиться через пакетування функцій, додаючи більше різноманітності до набору даних і зменшуючи кореляцію між деревами рішень. Залежно від типу проблеми визначення прогнозу буде різним. Для завдання регресії окремі дерева рішень будуть усереднені, а для завдання класифікації більшість голосів, тобто найпоширеніша категоріальна змінна, дасть прогнозований клас. Нарешті, зразок oob використовується для перехресної перевірки.

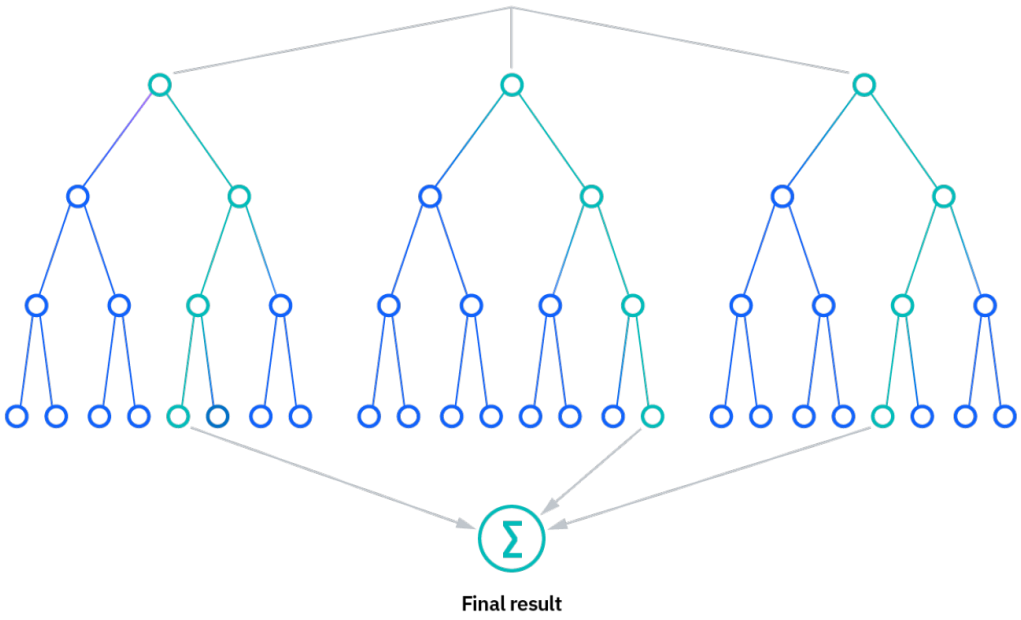

Рисунок 2.6 – Випадковий ліс

Алгоритм випадкового лісу має низку ключових переваг і проблем, коли він використовується для задач класифікації або регресії.

Переваги випадкового лісу:

– зменшення ризику переобладнання: дерева рішень мають ризик переобладнання, оскільки вони, як правило, щільно підходять для всіх зразків у навчальних даних. Однак, коли у випадковому лісі існує значна кількість дерев рішень, класифікатор не буде переповнювати модель, оскільки усереднення некорельованих дерев зменшує загальну дисперсію та помилку передбачення;

– забезпечує гнучкість: оскільки випадковий ліс може обробляти завдання як регресії, так і класифікації з високим ступенем точності, це популярний метод серед дослідників даних. Розміщення функцій також робить класифікатор випадкового лісу ефективним інструментом для оцінки відсутніх значень, оскільки він підтримує точність, коли частина даних відсутня;

– легко визначити важливість функції: випадковий ліс дозволяє легко оцінити важливість змінної або внесок у модель. Є кілька способів оцінити важливість функції. Важливість Джіні та середнє зменшення домішок (MDI) зазвичай використовуються для вимірювання того, наскільки зменшується точність моделі, коли дана змінна виключається. Однак важливість перестановки, також відома як точність зниження середнього значення (MDA), є ще одним показником важливості. MDA визначає середнє зниження точності шляхом випадкової перестановки значень ознак у зразках oob.

Недоліки випадкового лісу:

– процес займає багато часу: оскільки алгоритми випадкового лісу можуть обробляти великі набори даних, вони можуть надавати точніші прогнози, але можуть повільно обробляти дані, оскільки вони обчислюють дані для кожного окремого дерева рішень;

– вимагає більше ресурсів: оскільки випадкові ліси обробляють більші набори даних, їм знадобиться більше ресурсів для зберігання цих даних;

– більш складний: передбачення окремого дерева рішень легше інтерпретувати порівняно з лісом із них.

Алгоритм випадкового лісу застосовувався в багатьох галузях, що дозволило їм приймати кращі бізнес-рішення. Деякі випадки використання включають:

– фінанси: цьому алгоритму надається перевага над іншими, оскільки він скорочує час, витрачений на керування даними та завдання попередньої обробки. Його можна використовувати для оцінки клієнтів із високим кредитним ризиком, виявлення шахрайства та проблем із ціною опціонів;

– охорона здоров’я: алгоритм випадкового лісу має застосування в обчислювальній біології, що дозволяє лікарям вирішувати такі проблеми, як класифікація експресії генів, виявлення біомаркерів і анотація послідовностей. У результаті лікарі можуть оцінити реакцію на певні ліки;

– електронна комерція: її можна використовувати для механізмів рекомендацій для перехресних продажів.

2.2.7 k-nearest neighbor

Алгоритм k-nearest neighbor (k-найближчих сусідів), також відомий як KNN або k-NN, — це непараметричний класифікатор із контрольованим навчанням, який використовує близькість для класифікації або прогнозування щодо групування окремої точки даних. Хоча його можна використовувати як для регресії, так і для задач класифікації, зазвичай він використовується як алгоритм класифікації, виходячи з припущення, що подібні точки можна знайти поруч.

Для проблем класифікації мітка класу призначається на основі більшості голосів, тобто використовується мітка, яка найчастіше представлена навколо даної точки даних.

Рисунок 2.7 – k-найближчих сусідів

У задачах регресії використовується концепція, подібна до задачі класифікації, але в цьому випадку для прогнозування класифікації береться середнє k найближчих сусідів. Основна відмінність тут полягає в тому, що класифікація використовується для дискретних значень, тоді як регресія використовується для неперервних. Однак перш ніж можна буде зробити класифікацію, необхідно визначити відстань. Найчастіше використовується евклідова відстань. Варто також зазначити, що алгоритм KNN також є частиною сімейства моделей «ледачого навчання», тобто він зберігає лише навчальний набір даних, а не проходить етап навчання. Це також означає, що всі обчислення відбуваються під час класифікації або прогнозування. Оскільки для зберігання всіх навчальних даних він значною мірою покладається на пам’ять, його також називають методом навчання на основі екземплярів або пам’яті. Незважаючи на те, що він не такий популярний, як колись, це все ще один із перших алгоритмів, який вивчає наука про дані, завдяки його простоті та точності. Однак із зростанням набору даних KNN стає дедалі неефективнішим, що погіршує загальну продуктивність моделі.

Алгоритм k-NN використовувався в різних додатках, переважно в класифікації. Деякі з цих випадків використання включають:

– попередня обробка даних: набори даних часто містять відсутні значення, але алгоритм KNN може оцінити ці значення в процесі, відомому як імпутація відсутніх даних;

– механізми рекомендацій: використовуючи дані кліків із веб-сайтів, алгоритм KNN використовувався для надання автоматичних рекомендацій користувачам щодо додаткового вмісту. Це дослідження [16] показує, що користувача віднесено до певної групи, і на основі поведінки користувачів цієї групи їм надається рекомендація. Однак, враховуючи проблеми масштабування з KNN, цей підхід може бути не оптимальним для більших наборів даних;

– фінанси: він також використовувався в різних фінансових та економічних випадках використання;

– охорона здоров'я: KNN також знайшов застосування в галузі охорони здоров'я, роблячи прогнози щодо ризику серцевих нападів. Алгоритм працює, обчислюючи найімовірніші експресії генів;

– розпізнавання шаблонів: KNN також допомагає в ідентифікації шаблонів, наприклад у класифікації тексту та цифр. Це особливо корисно для визначення рукописних номерів, які ви можете знайти на формах або поштових конвертах.

Як і будь-який алгоритм машинного навчання, k-NN має свої сильні та слабкі сторони. Залежно від проекту та застосування, це може бути або не бути правильним вибором.

Переваги k-найближчих сусідів:

– легко реалізувати: враховуючи простоту й точність алгоритму, це один із перших класифікаторів, який вивчає новачок у дослідженні даних;

– легко адаптується: коли додаються нові навчальні зразки, алгоритм налаштовується з урахуванням будь-яких нових даних, оскільки всі навчальні дані зберігаються в пам’яті;

– кілька гіперпараметрів: KNN вимагає лише значення k і метрики відстані.

Недоліки k-найближчих сусідів:

– погано масштабується: оскільки KNN є ледачим алгоритмом, він займає більше пам’яті та зберігання даних порівняно з іншими

класифікаторами. Це може бути дорогим як з точки зору часу, так і з точки зору грошей. Більше пам’яті та сховища призведе до збільшення витрат бізнесу, а обчислення більшої кількості даних може зайняти більше часу. Хоча різні структури даних, такі як Ball-Tree, були створені для усунення неефективності обчислень, інший класифікатор може бути ідеальним залежно від бізнес-проблеми;

– проблема розмірності: алгоритм KNN погано працює з вхідними даними великої розмірності;

– схильність до переобладнання: через проблему розмірності KNN також більш схильний до переобладнання. Хоча методи вибору функцій і зменшення розмірності використовуються для запобігання цьому, значення k також може впливати на поведінку моделі. Нижчі значення k можуть перевищувати дані, тоді як вищі значення k мають тенденцію до «згладжування» прогнозованих значень. Однак, якщо значення k занадто велике, воно може не відповідати даним.

Висновки за розділом 2

Сучасній людині важко уявити своє життя без комп’ютеру, персональних пристроїв, смартфонів. Кожного дня ми користуємося засобами зв’язку (месенджерами, аудіо- або відео конференцій), пошуковими системами, соціальними мережами, веб-додатками тощо. Щоб отримати інформацію, необхідну для покращення рейтингу веб-сайту, запропонованого пошуковою системою, оцінка ефективності є важливою. Відповідно, у цьому розділі були розглянуті автоматизовані інструменти оцінки такі, як SEOptimer, Website Grader і Qualidator.

Але для задачі оцінки ефективності вебсайтів також є доречним використовувати машинне навчання.

Машинне навчання (ML) — це тип штучного інтелекту (AI), зосереджений на створенні комп’ютерних систем, які навчаються на

даних. Широкий спектр методів, які охоплює ML, дає змогу побудованим моделям покращувати свою продуктивність.

Тому, у цьому розділі також були проаналізовані основні принципи роботи таких методів машинного навчання, як linear regression, ridge regression, support vector machine, decision trees, random forest та k-nearest neighbour.

РОЗДІЛ 3
РОЗРОБКА МОДЕЛІ ОЦІНКИ ЕФЕКТИВНОСТІ ВЕБСАЙТІВ

3.1 Набір вхідних даних моделі

Для побудови будь-якої моделі машинного навчання необхідно спочатку формалізувати та обрати дані для тренування та тестування, щоб отримати результати моделювання та інтерпретувати їх.

3.1.1 Формування та формалізація вхідних даних

Для розробки моделі оцінки ефективності необхідно було обрати датасет для аналізу, який мав би всі необхідні дані та відповідав області використання результатів дослідження. Вибірка повинна мати унікальні дані та складатися із значень характеристик реальних сайтів.

Після аналізу багатьох існуючих наборів даних виявилось, що наразі мережа не налічує безкоштовних та відповідних до вимог датасетів.

Тому для аналізу було створено власний датасет, в якому відібрано ті характеристики вебсайтів, за якими можна оцінити ефективність роботи веб-сайтів. Ці показники були взяті з веб-ресурсу similar.web, на якому є можливість проаналізувати будь-який веб-сайт та зібрати дані для подальшої роботи. З отриманих даних за допомогою Microsoft Excel було сформовано вибірку, яка наразі налічує 101 рядок.

Сам набір даних містить 9 параметрів (рис. 3.1):

- Domain – доменне ім’я вебсайту;
- Category – певна категорія, до якої відноситься вебсайт;
- Monthly visit – кількість відвідувачів на сайті за місяць;
- Visit duration – середня тривалість візиту користувача на сайті;
- Pages – середня кількість сторінок, які переглядає користувач за один візит на сайті;

– Bounce rate (%) – відсоток відвідувачів, які переглядають лише одну сторінку перш ніж покинути його;

– Social traffic (%) – відсоток соціальних мереж, які спрямували трафік на певний сайт;

– Display Advertising (%) – відсоток медійної реклами вебсайту;

– Estimation – оцінка сайту в певній категорії. Значення Estimation було отримано з формули

$$Estimation = (total\ number - rank)/total\ number, \quad (3.1)$$

де *total number* – загальна кількість сайтів за кожною категорією, а *rank* – ранг сайту.

	Domain	Category	Monthly visit	\
0	google.com	Search_Engines	25.167798	
1	facebook.com	Social_Networks_and_Online_Community	23.558242	
2	twitter.com	Social_Networks_and_Online_Community	22.559521	
3	instagram.com	Social_Networks_and_Online_Community	22.616830	
4	tiktok.com	Social_Networks_and_Online_Community	21.531971	

	Visit duration	Pages	Bounce rate (%)	Social traffic (%)	\
0	10.39	8.69	0.2853	0.0191	
1	10.33	8.70	0.3133	0.0184	
2	10.39	9.88	0.3205	0.0346	
3	8.14	11.01	0.3515	0.0607	
4	3.47	7.58	0.3669	0.0794	

	Display Advertising (%)	Estimation
0	0.005	0.990
1	0.005	0.997
2	0.005	0.995
3	0.005	0.996
4	0.005	0.986

Рисунок 3.1 – Приклад перших 5 записів сформованого датасету

3.1.2 Попередня обробка даних

Попередня обробка даних відіграє велику роль у моделюванні. Правильно підготовлені дані покращують роботу будь-якої моделі та не допускають неправильного процесу моделювання.

Оскільки датасет мав завеликі значення, значення були масштабовані для зручності використання та аналізу даних.

Далі потрібно визначити: чи потрібно проводити нормалізацію даних. Нормалізація – це процес приведення таблиць бази даних до ряду нормальних форм з метою уникнення надлишковості в базі даних, аномалій вставки, редагування та видалення даних.

Надлишковість даних виникає при неправильному проектуванні таблиці бази даних. У цьому випадку таблиця містить групи даних, що повторюються. Такі групи даних виникають, коли здійснюється спроба записати в одну комірку таблиці більше одного значення.

Аномалія вставки – це додавання небажаної або неіснуючої (вигаданої) інформації про одну сутність в момент вставки інформації про іншу сутність.

Аномалія редагування – це вимушена необхідність зміни (оновлення) даних у всій таблиці у випадку їх зміни (оновлення) в одній комірці таблиці з метою уникнення їх двоякого трактування.

Аномалія видалення – це втрата одних даних в таблиці при видаленні інших даних в таблиці.

Тож, судячи з того, що датасет не має надлишковості даних або аномалій модифікацій в базі даних (аномалії вставки, редагування та видалення даних), проводити нормалізацію даних не потрібно.

3.1.3 Вибір параметрів веб-сайтів для побудови моделі

Після обробки даних постає питання: чи потрібно використовувати всі параметри для моделювання. Оскільки в сформованій вибірці є параметри, які не мають числових значень, то доцільним є не використовувати їх для подальшого моделювання.

Тож, для побудови моделі потрібно розділити дані на data і target, де target – це цільова ознака, а data – залежні змінні. У якості цільової ознаки буде значення Estimation, а у якості залежних змінних – усі параметри, окрім тих, які не мають числових значень, а саме Domain та Category, які містять текстові дані.

3.2 Вибір програмного комплексу для побудови моделі

Для розробки моделі використовується мова програмування Python з використанням зовнішніх бібліотек для статистичного аналізу даних, а також для побудови моделей прогнозування.

У роботі були використані бібліотеки та методи: numpy, pandas, matplotlib.pyplot, train_test_split, metrics, LinearRegression, Ridge, GridSearchCV, RandomForestRegressor, GradientBoostingRegressor, SVR.

Пакети pandas та numpy забезпечують швидкі, гнучкі та виразні структури даних, розроблені для того, щоб зробити роботу з «реляційними» або «міченими» даними одночасно легкою та інтуїтивно зрозумілою. Вони мають на меті стати основним будівельним блоком високого рівня для практичного аналізу реальних даних у Python.

Пакет matplotlib.pyplot це інтерфейс на основі matplotlib. Він забезпечує неявний, схожий на MATLAB, спосіб побудови. Він також відкриває фігури на екрані та діє як менеджер графічного інтерфейсу користувача.

Пакет train_test_split – розбиття масивів або матриць на випадкові навчальні та тестові вибірки.

Модуль metrics реалізує функції оцінки похибки передбачення для конкретних цілей [17].

LinearRegression підходить до лінійної моделі з коефіцієнтами w = (w1, …, wp), щоб мінімізувати залишкову суму квадратів між спостережуваними цілями в наборі даних і цілями, передбаченими лінійним наближенням [17].

Ridge модель розв’язує регресійну модель, де функція втрат є лінійною функцією найменших квадратів, а регулярізація задана l2-нормою [17].

GridSearchCV – вичерпний пошук за вказаними значеннями параметрів для оцінювача [17].

RandomForesttRegressor – це метаоцінювач, який підходить для ряду класифікаційних дерев рішень на різних підвибірках набору даних і

використовує усереднення для підвищення точності прогнозування та контролю надпідгонки [17].

GradientBoostingRegressor будує адитивну модель поетапно; це дозволяє оптимізувати довільні диференційовані функції втрат. На кожному етапі дерево регресії підбирається на від'ємному градієнті заданої функції втрат [17].

SVR будує модель, яка залежить лише від підмножини навчальних даних, оскільки функція вартості ігнорує вибірки, прогноз яких близький до цільового [17].

Увесь перелічений набір бібліотек дозволяє швидко побудувати модель оцінки ефективності веб-сайтів та забезпечити вхідні дані моделі необхідною попередньою обробкою, що дозволяє отримати кращі результати моделювання.

3.3 Метрики оцінки моделей регресії

Чотири загальні метрики оцінювання проблем регресії – це середня абсолютна похибка (MAE), середня квадратична помилка (MSE), середньоквадратична помилка (RMSE), R квадрат (R^2).

3.3.1 Середня абсолютна похибка

Середня абсолютна похибка (MAE) — це середнє абсолютне значення похибок:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{3.2}$$

де n – кількість спостережень у наборі даних, y_i – справжнє значення, $\hat{y}_i$ – прогнозоване значення.

MAE – це лінійна оцінка, що означає, що всі індивідуальні відмінності однаково впливають на середнє значення. Він забезпечує оцінку розміру похибки, але не її напрямок (наприклад, завищення чи заниження прогнозу).

Середня абсолютна похибка (MAE) є вирішальним статистичним показником для регресійних моделей, оскільки це простий для розуміння,

інтерпретований і надійний інструмент для оцінки точності прогнозів. Серед багатьох причин найбільш значущими є:

– стійкість до викидів. Екстремальні результати не так сильно впливають на MAE, як на інші показники, такі як середня квадратична помилка (MSE). Це робить його відповідним показником для наборів даних, які містять викиди або екстремальні значення;

– лінійна оцінка. Усім індивідуальним відмінностям надається однакова вага в середньому. Це спрощує порівняння продуктивності кількох моделей або варіантів однієї моделі;

– інтерпретація MAE є основною та очевидною статистикою, яка представляє середню величину помилок прогнозів. Це просто для розуміння нетехнічним зацікавленим сторонам;

– ті самі одиниці, що й змінна відповіді. MAE виражається в тих самих одиницях, що й змінна відповіді, що дозволяє легко зрозуміти розмір помилки передбачення. Це важливо під час спроби зрозуміти продуктивність моделі в контексті проблеми, яку вирішують;

– використовується в кількох дисциплінах. MAE використовується у фінансах, техніці та метеорології. У деяких випадках це навіть вважається стандартним заходом;

– пропонує інформацію про розмір помилки. MAE надає інформацію про величину помилки, створеної моделлю. Це дозволяє порівнювати моделі та вибирати найкращу, а також покращувати модель шляхом визначення прогнозованої середньої абсолютної відсоткової похибки.

Підсумовуючи, MAE є широко використовуваним і важливим показником продуктивності для регресійних моделей, оскільки він інтуїтивно зрозумілий, інтерпретований, стійкий до викидів і пропонує інформацію про розмір помилки.

3.3.2 Середня квадратична помилка

Середня квадратична помилка (MSE) – це середнє квадратичних помилок:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad (3.3)$$

де n – кількість повторень, y_i – справжнє значення, $\hat{y}_i$ – прогнозоване значення.

Оцінки середньоквадратичної помилки досить близькі до обчислень дисперсії.

MSE – це середня різниця між спостережуваними та очікуваними значеннями, виміряними в квадратних одиницях. Використання квадратичних одиниць замість одиниць природних даних робить інтерпретацію менш очевидною. Мета зведення розбіжностей у квадрат є багатогранною.

Зведення різниць у квадрат усуває різницю негативних середніх квадратів помилок і гарантує, що квадрат середньої помилки завжди більший або дорівнює нулю. Значення зазвичай завжди позитивне. Лише модель без будь-яких помилок матиме нульовий MSE. Цього насправді не відбувається. Крім того, зведення у квадрат посилює ефект більшої неточності. Ці обчислення карають більші помилки непропорційно більше, ніж менші. Цей атрибут необхідний, якщо ви хочете, щоб помилок вашої моделі було менше.

Середня квадратична помилка є мірою точності, яка використовує натуральні одиниці даних.

3.3.3 Середньоквадратична помилка

Середньоквадратична помилка (RMSE) – це квадратний корінь із середнього квадратичного значення помилок:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \quad (3.4)$$

де n – кількість спостережень у наборі даних, y_i – справжнє значення, $\hat{y}_i$ – прогнозоване значення.

RMSE – це показник, який переважно використовується в регресійному аналізі та прогнозуванні, де точність має велике значення. Чим нижче RMSE, тим краща здатність моделі точно прогнозувати. І навпаки, вищий RMSE означає більшу розбіжність між прогнозованими та фактичними результатами.

RMSE зазвичай використовується в машинному навчанні, оскільки він надає відносно велику вагу великим помилкам. Це означає, що RMSE має бути більш корисним, коли великі помилки особливо небажані. Він також цінний, оскільки він зберігає ті самі одиниці, що й вхідні дані, що полегшує його інтерпретацію.

Однак, незважаючи на численні переваги, важливо пам’ятати, що RMSE не є єдиним показником точності моделі, і він має свої обмеження. Наприклад, RMSE не повідомляє нам, наскільки добре працюватиме майбутня модель або чи найкраще вона підходить для даних. Це найбільш корисно, коли використовується разом з іншими показниками, як-от середня абсолютна похибка (MAE), щоб отримати більш повне уявлення про ефективність моделі.

Підсумовуючи, середньоквадратична помилка служить фундаментальною опорою в царині статистичного аналізу та машинного навчання, пропонуючи спрощену, але ефективну міру похибки передбачення. Незважаючи на свої обмеження, за належного використання та в поєднанні з іншими відповідними показниками RMSE може надати суттєве розуміння ефективності та надійності прогнозних моделей. Тому розуміння та належне використання цього показника має вирішальне значення для всіх, хто займається аналізом даних або прогнозуванням моделей, підвищуючи точність і ефективність своєї роботи.

3.3.4 R квадрат

R Squared (R^2) – коефіцієнт детермінації.

R^2 – це статистичний показник дисперсії в прогнозах моделі, який підтверджує відповідність прогнозованих значень фактичним значенням.

$$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \tag{3.5}$$

де n – кількість спостережень у наборі даних, y_i – справжнє значення, $\hat{y}_i$ – прогнозоване значення.

R-квадрат вказує на ідеальну продуктивність моделі, коли він дорівнює 1, і на погану, коли він дорівнює 0. Чим ближче значення r-квадрата до 1, тим краще підходить модель.

3.4 Навчання моделі

Для того, щоб навчити модель потрібно розділити дані на навчальні та тестові вибірки та оцінити роботу отриманих наборів даних за допомогою методу score.

3.4.1 Навчальна та тестова вибірки

У машинному навчанні набори даних зазвичай поділяються на дві підмножини: навчальні та тестові дані. Навчальні дані використовуються для навчання алгоритму машинного навчання. Дані тестування використовуються для оцінки точності навченого алгоритму.

Дані тестування можна використовувати для оцінки точності, надійності та справедливості моделі. Його також можна використовувати для визначення областей, де модель потребує вдосконалення.

Навчальні дані використовуються, щоб навчити модель машинного навчання робити прогнози або виконувати бажане завдання. Зазвичай він позначений, що означає, що вихідні дані моделі відомі для кожної точки даних. Модель вчиться визначати шаблони в даних і робити прогнози на основі цих шаблонів.

Дані тестування використовуються для оцінки продуктивності моделі машинного навчання. Зазвичай вони відрізняються від даних навчання та не позначаються. Це означає, що вихідні дані моделі невідомі для кожної точки даних. Оцінюється здатність моделі робити точні прогнози на основі даних тестування.

3.4.2. Метод score

Значення score повертає коефіцієнт детермінації прогнозу. Коефіцієнт детермінації визначається як

$$(1 - \frac{u}{v}), \tag{3.6}$$

де *u* – це залишкова сума квадратів ((y_true - y_pred)** 2).sum(); *v* – це загальна сума квадратів ((y_true - y_true.mean()) ** 2).sum().

Найкраща можлива оцінка – 1,0 і вона може бути негативною (оскільки модель може бути як завгодно гіршою). Константна модель, яка завжди передбачає очікуване значення y, не враховуючи вхідні характеристики, отримає оцінку 0,0.

3.4.3 Функція GreadSearchCV

Метою є удосконалити модель будь-яким можливим способом. Одним з важливих факторів ефективності моделей є їхні гіперпараметри. Як тільки ми встановимо відповідні значення для цих гіперпараметрів, продуктивність моделі може значно покращитися.

Гіперпараметри – це параметри, які явно задані та керують процесом навчання.

GridSearchCV – це процес налаштування гіперпараметрів для визначення оптимальних значень для даної моделі. Виконання цього вручну може зайняти значну кількість часу та ресурсів, тому ми використовуємо GridSearchCV для автоматизації налаштування гіперпараметрів.

GridSearchCV – це функція, яка постачається в пакеті Scikit-learn (або SK-learn) model_selection.

Грід-пошук – це метод для виконання гіперпараметричної оптимізації, тобто з заданою моделлю і тестовим набором даних, оскільки тестова вибірки використовується для оцінки ефективності моделі, це метод для знаходження оптимальної комбінації гіперпараметрів. Мета полягає в тому, щоб навчити та оцінити кожну з цих моделей. Потім обрати той, який показав себе найкраще.

3.5 Результати дослідження

У роботу навчені різні моделі оцінки ефективності на наборі даних і буде обрана для практичного застосування та, яка має найкращу продуктивність. Однак є можливості для вдосконалення, оскільки ми не можемо точно сказати, що ця конкретна модель найкраще підходить для вирішення проблеми.

3.5.1 Результати оцінки роботи моделей

Після того, як ми імпортували бібліотеки, розділили дані на навчальний та тестовий набори і обрали метрики оцінки регресії, ми оцінили роботу моделі за допомогою п’ятьох методів регресії: лінійна регресія, рідж регресія, випадковий ліс, градієнтний бустінг та машина опорних векторів. Результати цього дослідження на рис. 3.2.

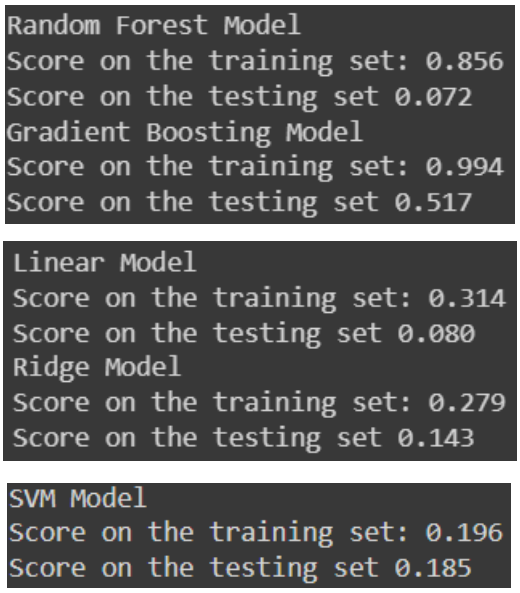

Рисунок 3.2 – Оцінка роботи моделей

Отже, були отримані значення score для навчальної та тестової вибірки за п’ятьма методами. Для подальшого дослідження були відібрані три моделі, які

дали кращу оцінку за тестовим набором даних: Ridge Model – 0.143; Gradient Boosting Model – 0.484; SVM Model – 0.185.

3.5.2 Результати покращення ефективності моделі

За допомогою функції GreadSearchCV спершу був проведений тест для моделі Ridge Model, яка має гіперпараметр alpha.

По результатам тесту було визначено, що оптимальне значення alpha – 2 (рис. 3.3).

```
Fitting 5 folds for each of 4 candidates, totalling 20 fits
Best parameters are  {'alpha': 2}
Best score is  -2.206865681378912
```

Рисунок 3.3 – Результат визначення оптимального гіперпараметру для Ridge Model

Після визначення alpha було зроблено перевірку щодо покращення ефективності моделі. Так як, до пошуку оптимального гіперпараметру Ridge Model мала значення score для тестового набору даних – 0.143, а після застосування alpha = 2 – 0.164 (рис. 3.4), можна зробити висновок щодо покращення ефективності моделі.

```
Score on the training set: 0.260
Score on the testing set 0.164
```

Рисунок 3.4 – Результат оцінки роботи моделі після використання гіперпараметру

Другим був проведений тест для моделі Gradient Boosting Model, яка має гіперпараметри learning_rate та max_depth.

По результатам тесту було визначено, що оптимальне значення learning_rate – 0.5 та max_depth – 3 (рис. 3.5).

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
Best parameters are  {'learning_rate': 0.5, 'max_depth': 3}
Best score is  -2.355619773870792
```

Рисунок 3.5 – Результат визначення оптимальних гіперпараметрів для Gradient Boosting Model

Після визначення learning_rate та max_depth було зроблено перевірку щодо покращення ефективності моделі. Так як, до пошуку оптимальних гіперпараметрів Gradient Boosting Model мала значення score для тестового набору даних – 0.484, а після застосування learning_rate – 0.5 та max_depth – 3 – 0.560 (рис. 3.6), можна зробити висновок щодо покращення ефективності моделі.

```
Score on the training set: 1.000
Score on the testing set 0.560
```

Рисунок 3.6 – Результат оцінки роботи моделі після використання гіперпараметрів

Також був проведений тест для моделі SVM Model, яка має гіперпараметри C та gamma.

По результатам тесту було визначено, що оптимальне значення C – 1000 та gamma – 0.0001 (рис. 3.7).

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
Best parameters are  {'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'}
Best score is  -0.9991860326882807
```

Рисунок 3.7 – Результат визначення оптимальних гіперпараметрів для SVM Model

Після визначення C та gamma було зроблено перевірку щодо покращення ефективності моделі. Так як, до пошуку оптимальних гіперпараметрів SVM Model мала значення score для тестового набору даних – 0.185, а після

застосування C – 1000 та gamma – 0.0001 – 200 (рис. 3.8), можна зробити висновок щодо покращення ефективності моделі.

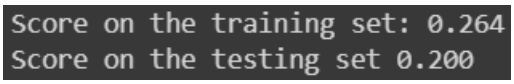

Рисунок 3.8 – Результат оцінки роботи моделі після використання гіперпараметрів

Отже, для кожної моделі за допомогою GreadSearchCV було визначено найоптимальніші гіперпараметри для покращення продуктивності моделей.

3.5.3 Результати оцінки роботи моделей

Після знаходження оптимальні гіперпараметри для кожної моделі, було розраховано середня абсолютна похибка (MAE), середня квадратична помилка (MSE), середньоквадратична помилка (RMSE), R квадрат (R^2).

На рис. 3.9 результати розрахунку кожної метрики та графік цільового і прогнозованого значення Estimation для моделі Ridge Model.

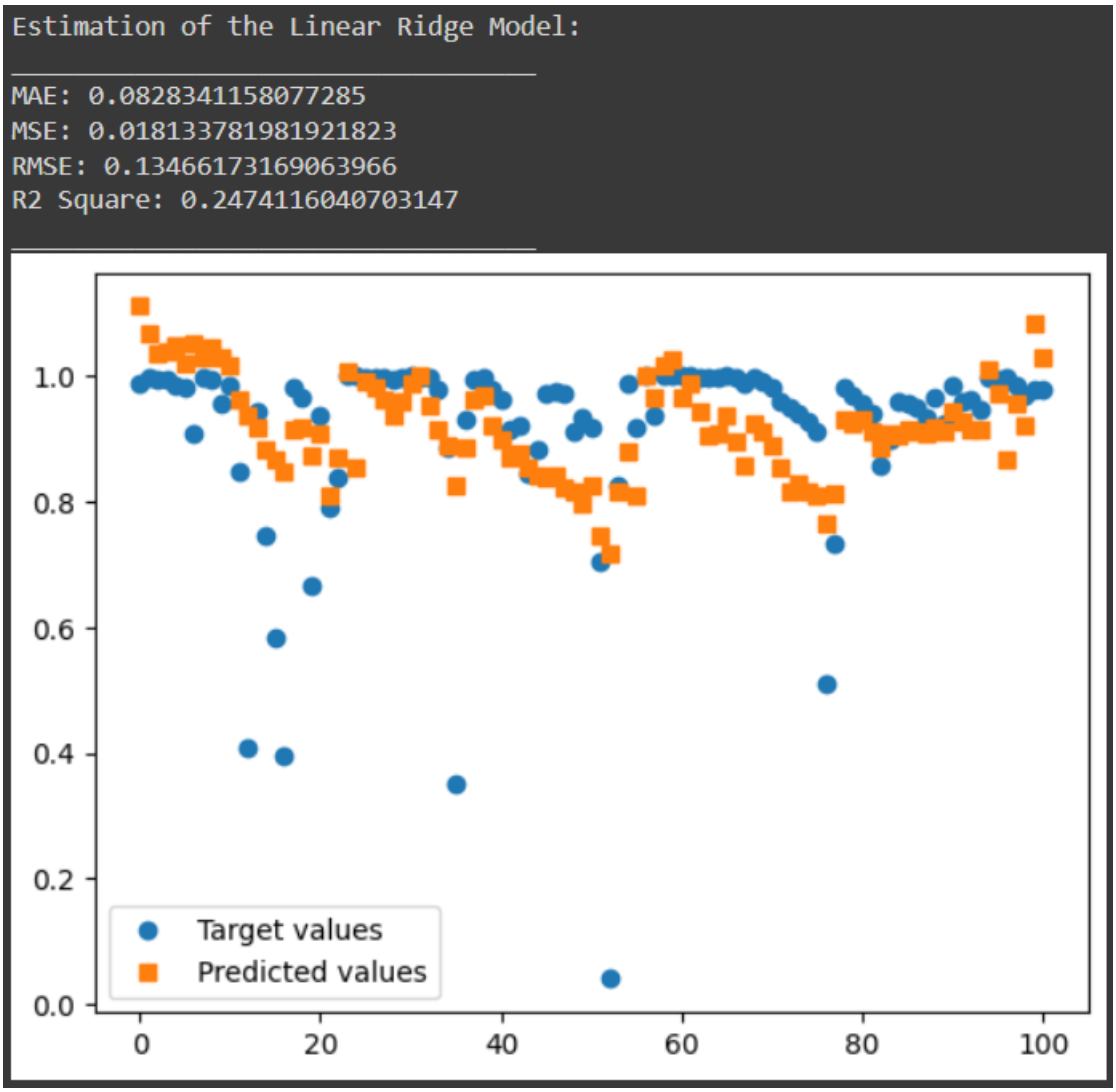

Рисунок 3.9 – Результати розрахунку метрик та графік цільового і прогнозованого значення Estimation для моделі Ridge Model

На рис. 3.10 результати розрахунку кожної метрики та графік цільового і прогнозованого значення Estimation для моделі Gradient Boosting Model.

```
Estimation of the Gradient Boosting Model:
_____________________________________
MAE: 0.012029300818788537
MSE: 0.0023101788960496355
RMSE: 0.04806432040557357
R2 Square: 0.9041229330196046
_____________________________________
<ipython-input-35-47f3c8aeddf4>:13: FutureWarning: The frame.append meth
  results_df = results_df.append(results_df_1, ignore_index=True)
```

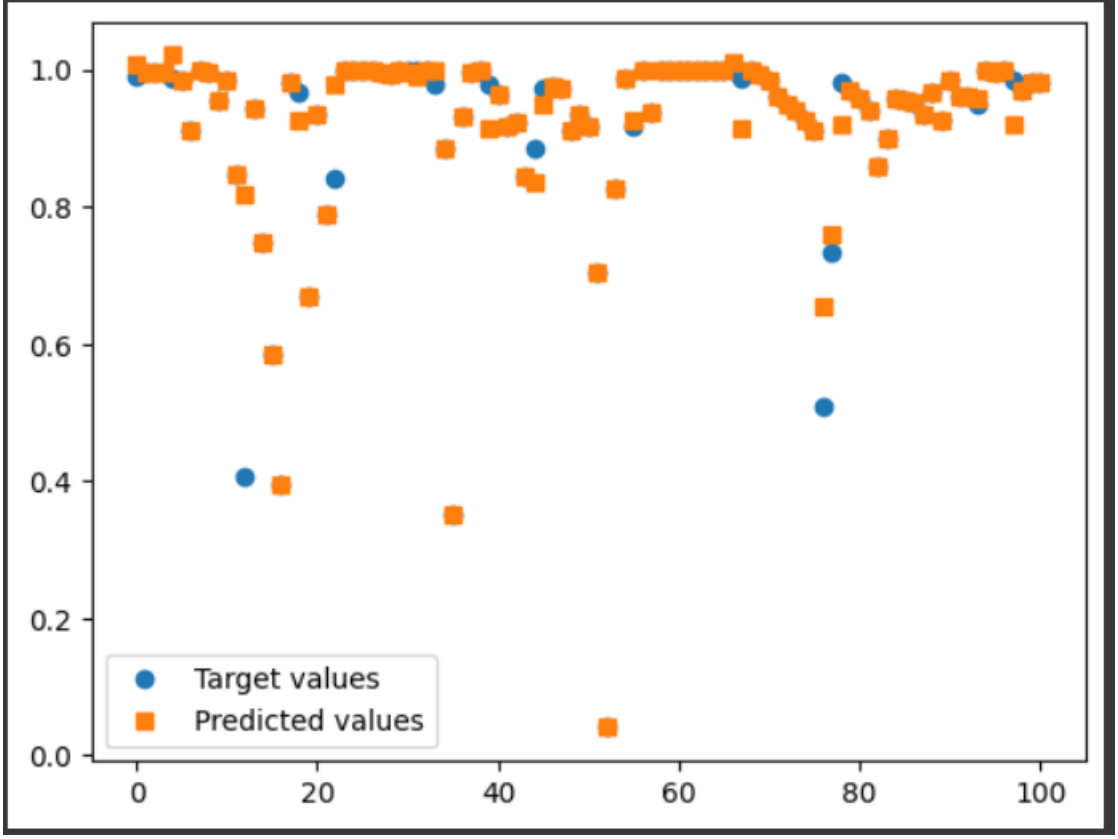

Рисунок 3.10 – Результати розрахунку метрик та графік цільового і прогнозованого значення Estimation для моделі Gradient Boosting Model

На рис. 3.11 результати розрахунку кожної метрики та графік цільового і прогнозованого значення Estimation для моделі SVM Model.

```
Estimation of the SVM Model:
_____________________________________
MAE: 0.08916091375817967
MSE: 0.01940279740575233
RMSE: 0.1392939245112734
R2 Square: 0.19474491362577584
_____________________________________
<ipython-input-39-bf0f31b9c487>:13: FutureWarning: The frame.append meth
  results_df = results_df.append(results_df_2, ignore_index=True)
```

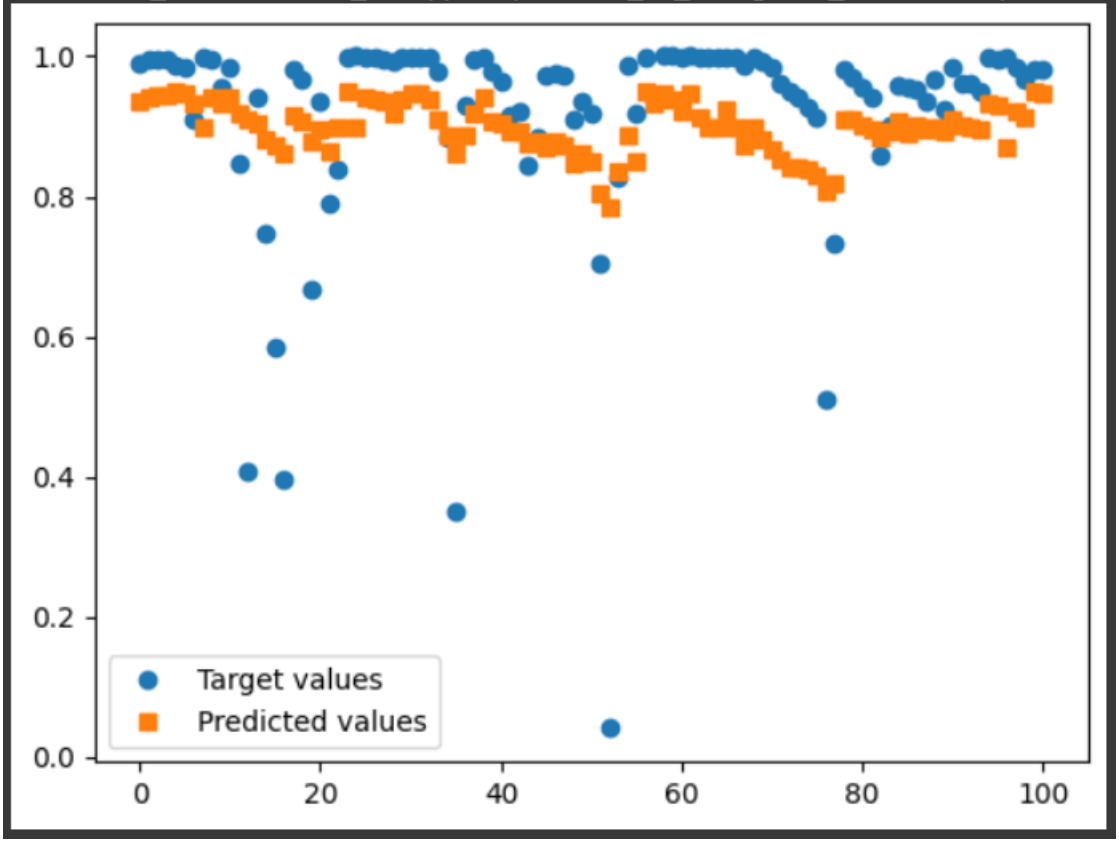

Рисунок 3.11 – Результати розрахунку метрик та графік цільового і прогнозованого значення Estimation для моделі SVM Model

На рис. 3.12 показані значення усіх метрик для кожної моделі для порівняння результатів.

	Model	MAE	MSE	RMSE	R2 Square
0	Linear Ridge Regression	0.082834	0.018134	0.134662	0.247412
1	Gradient Boosting Regressor	0.012029	0.002310	0.048064	0.904123
2	SVM Regressor	0.089161	0.019403	0.139294	0.194745

Рисунок 3.12 – Результати метрик для кожної моделі

Оскільки показники MAE, MSE, RMSE – це метрики помилок, їх значення повинно прагнути 0. Показник R2 Square підтверджує відповідність прогнозованих даних з фактичними, то його значення повинно прагнути 1. Також відповідність прогнозованих та фактичних даних показують побудовані графіки.

Тому, дивлячись на отримані дані, можна зробити висновок, що модель Gradient Boosting Regressor показала найкращий результат.

Висновки за розділом 3

У цьому розділі було описано створення датасету, навчання моделі та результати дослідження.

Створений датасет має 101 рядок та 9 параметрів, оснований на даних реальних веб-сайтів.

Дані навчання та дані тестування є найважливішими для розуміння для створення точних і надійних моделей машинного навчання. Ретельно відбираючи та готуючи дані, можна покращити продуктивність моделей і переконатися, що вони готові до використання в реальних умовах.

У дослідженні було застосовано три методи: рідж регресію, регресію градієнтного бустінгу та регресію машини опорних векторів. Також знайдено оптимальні гіперпараметри, за допомогою них покращено ефективність моделей і розраховано метрики для їх оцінки.

Експерименти були застосовані на наборі даних, який було зібрано, щоб отримати найкращу ефективність методів регресії для розробки нової динамічної моделі для оцінки ефективності веб-сайтів.

У результаті були отримані найкращі значення для моделі Gradient Boosting Regressor: MAE – 0.01; MSE – 0.002; RMSE – 0.4; R2 Square – 0.9.

Таким чином, побудовану модель оцінки ефективності веб-сайтів можна застосовувати для надання оцінок роботи веб-сайтів.

ВИСНОВКИ

У сучасному світі важко уявити своє життя без смартфонів, комп’ютерів та інших девайсів. За допомогою них кожен день люди користуються месенджерами, пошуковими система, дивлять відео та фільми, читають книжки та статті і навіть замовляють ті чи інші товари або продукти. Інтернет-технології використовують майже усі сфери повсякденного життя. Тому перед розробниками стоїть задача вдосконалення ефективності роботи додатків та вебсайтів для долучення більшої кількості користувачів.

У даній роботі було проаналізовано роботу інтернет-ресурсів, їх структуру, функціональність, технічні вимоги, вимоги користувачів та критерії оцінки ефективності вебсайтів. Також були розглянуті існуючи автоматизовані системи та методи машинного навчання для оцінки продуктивності вебсайтів.

Однією з задач було створення датасету, використовуючи характеристики реальних сайтів та створення методу оцінки ефективності вебсайтів за допомогою за допомогою машинного навчання. На основі цього, було створено такі регресійні моделі, як Ridge Regression Model, Support Vector Machine Regression Model та Gradient Boosting Regression Model. Далі за допомогою функції GreadSearchCV для кожної моделі були отримані значення гіперпараметрів, із застосуванням яких були покращені оцінки навчання та тестування кожної з моделей. Після отриманих результатів було розраховано значення чотирьох загальних метрик оцінювання проблем регресії: середня абсолютна похибка (MAE), середня квадратична помилка (MSE), середньоквадратична помилка (RMSE), R квадрат (R^2) для трьох даних моделей та порівняно їх. Експерименти були застосовані на наборі даних, який було зібрано, щоб отримати найкращу ефективність методів регресії для розробки нової динамічної моделі для оцінки ефективності вебсайтів.

У результаті чого було отримано метод оцінки ефективності вебсайтів з динамічною регресійною моделлю градієнтного бустінгу та використанням чотирьох метрик оцінювання, таких як MAE, MSE, RMSE, R^2.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. J. Y. Lemos, and A. R. Joshi, "Search engine optimization to enhance user interaction," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud) (I-SMAC), Palladam, 2017, pp. 398-402. [Електронний ресурс] Режим доступу: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8058379&isnumber=805824;

2. D. Sharma, R. Shukla, A. K. Giri and S. Kumar, "A Brief Review on Search Engine Optimization," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 687-691;

3. Megan Hendrickson “How to optimize use website for mobile devices”. [Електронний ресурс] Режим доступу: https://www.dreamhost.com/blog/how-to-optimizeyoursite-formobile/;

4. S. Gupta, N. Rakesh, A. Thakral and D. K. Chaudhary, "Search engine optimization: Success factors," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, 2016, pp. 17-21. DOI: 10.1109/PDGC.2016.7913146. [Електронний ресурс] Режим доступу: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7913146&isnumber=791317;

5. M. C. Anchahua, L. V. Garnique, and J. A. Tarazona, "User Experience Maturity Model for Ecommerce Websites," 2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI), Bogota, 2018, pp. 1-6;

6. F. Paz and J. A. Pow-Sang, "Usability Evaluation Methods for Software Development: A Systematic Mapping Review," 2015 8th International

Conference on Advanced Software Engineering & Its Applications (ASEA), Jeju, 2015, pp. 1-4;

7. M. Bilal, Z. Yu, S. Song, and C. Wang, "Evaluate Accessibility and

Usability Issues of Particular China and Pakistan Government Websites," 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 2019, pp. 316-322;

8. Website Grader, [Електронний ресурс] Режим доступу: https://website.grader.com/;

9. Seoptimer, [Електронний ресурс] Режим доступу: https://www.seoptimer.com/;

10. Z. Zhu, X. Fei, and Y. Guizhou, "Research on Performance Optimization for the Web-Based University Educational Management Information System," 2011 International Conference on Intelligence Science and Information Engineering, Wuhan, 2011, pp. 261-264;

11. D. Sharma, R. Shukla, A. K. Giri and S. Kumar, "A Brief Review on Search Engine Optimization," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 687-69;

12. Singh, K.K.; Kumar, P.; Mathur, J. Implementation of a Model for Websites Quality Evaluation—DU Website. Int. J. Innov. Adv. Comput. Sci. 2014, 3, 27–37

13. Vapnik, V. The Nature of Statistical Learning Theory. Springer, New York, 1995;

14. Support Vector Machine – Regression (SVR), [Електронний ресурс] Режим доступу: http://www.saedsayad.com/support_vector_machine_reg.htm;

15. Graczyk, M., Lasota, T., & Trawiński, B. (2009, October). Comparative analysis of premises valuation models using KEEL, RapidMiner, and WEKA. In International conference on computational collective intelligence (pp. 800-812). Springer, Berlin, Heidelberg;

16. Електронний ресурс. Режим доступу: https://www.researchgate.net/publication/267572060_Automated_Web_Usage_Data_Mining_and_Recommendation_System_using_K-Nearest_Neighbor_KNN_Classification_Method;

17. Електронний ресурс. Режим доступу: https://scikit-learn.org/;

18. Qu, Q.X., Guo, F. & Duffy, V.G. (2017) Effective use of human physiological metrics to evaluate website usability: An empirical investigation from China. Aslib Journal of Information Management. Vol. 69, No. 4, p370-388;

19. University of Leicester (2017) ‘Evaluating websites‘. [Електронний ресурс] Режим доступа: https://www2.le.ac.uk/offices/ld/resources/study/eval-web;

20. University of Brighton (2017) ‘Brighton Business School’. [Електронний ресурс] Режим доступа: https://www.brighton.ac.uk/bbs/index.aspx;

21. Open Business Council (2017) ‘What Does Your Website Say About You?’. [Електронний ресурс] Режим доступа: https://www.openbusinesscouncil.org/2017/08/what-does-your-website-say-about-you/;

22. Newbold, C. (2014) ‘How to Evaluate a Website’. [Електронний ресурс] Режим доступа: http://thevisualcommunicationguy.com/2014/08/27/how-to-evaluate-a-website/;

23. Hernandez, A. & Resnick, M. L. (2013) Placement of Call to Action Buttons for Higher Website Conversion and Acquisition An Eye Tracking Study. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 57, No. 1, p1042-1046. SAGE Publications;

24. Bauer, C. & Scharl, A. (2000) Quantitative evaluation of Web site content and structure. Internet Research. Vol. 10, No. 1, p31-44;

25. Bluford Library (2017) ‘Evaluating Web Resources’. [Електронний ресурс] Режим доступа: http://libguides.library.ncat.edu/content.php?pid=53820&sid=394505.

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Магістр**
Галузь знань: **15 – Автоматизація та приладобудування**
Спеціальність: **151 – «Автоматизація та комп'ютерно-інтегровані технології»**

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної та прикладної системотехніки

_____**д.т.н., проф. Шматков С. І.**

«08 » грудня 2022 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ)

Чепак Анастасія Альбертівна
(прізвище, ім'я, по батькові студента)

1. Тема роботи «Метод оцінки ефективності веб-сайтів»

керівник роботи Стрілець Вікторія Євгеніївна, канд. техн. наук, доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «10» листопада 2023 року № 4101-5/3197

2. Строк подання студентом роботи 28 .11.2023

3. Перелік питань, які потрібно розробити:
1. Аналіз існуючих підходів та методів для оцінки ефективності веб-сайтів.
2. Розробка методу оцінки ефективності веб-сайту на основі алгоритмів машинного навчання.
3. Програмна реалізація методу оцінки ефективності веб-сайту з використанням даних, які описують роботу реального веб-сайту, та її тестування.
4. Надання рекомендації щодо застосування розробленого методу та напрямків його вдосконалення.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1.	Аналіз наукової та профільної літератури з підходів та методології оцінки ефективності роботи веб-сайтів;	08.12.2022 – 01.02.2023
2.	Порівняння та аналіз існуючих методів оцінки ефективності веб-сайтів;	01.02.2023 – 15.03.2023
3.	Дослідження можливостей впровадження алгоритмів машинного навчання для удосконалення методів оцінки ефективності веб-сайтів;	15.03.2023 – 30.04.2023
4.	Розробка методу оцінки ефективності веб-сайтів;	30.04.2023 – 01.07.2023
5.	Програмна реалізація і тестування методу оцінки ефективності роботи веб-сайтів на реальному наборі даних;	01.07.2023 – 30.09.2023
6.	Дослідження основних показників якості роботи методу та можливостей їх покращення;	30.09.2023 – 30.10.2023
7.	Аналіз отриманих результатів та висновки;	30.10.2023 – 28.11.2023
8.	Представлення кваліфікаційної роботи.	04.12.2023

5. Дата видачі завдання 08.12.2022р.

Студент Чепак А.А. _____
ініціали, прізвище підпис

Керівник роботи Стрілець В.Є. _____
ініціали, прізвище підпис

Додаток Б

Технічне завдання

на розробку програмного виробу «Метод оцінки ефективності вебсайтів»

Назва розділу	Назва і зміст підрозділу
1. Введення	1.1. Назва програмного виробу: Метод оцінки ефективності вебсайтів. 1.2. Галузь застосування: програмна реалізація методів статистичного аналізу.
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології». 2.2. Завдання на кваліфікаційну роботу магістра затверджено наказом ректора № _____ від _____.
3.Призначення розробки	3.1. Мета розробки програмного виробу: розробка моделі для оцінки продуктивності веб-ресурсів та її подальше оцінювання для можливості покращення показників ефективності. 3.2. Призначення програмного виробу: побудовану модель оцінки ефективності вебсайтів можна застосовувати для надання оцінок роботи вебсайтів. 3.3. Вихідні дані для розробки: дані засновані на характеристиках реальних вебсайтів.
4. Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: можливість оцінки ефективності вебсайтів з використанням методів машинного навчання. 4.2. Вимоги до надійності: забезпечення працездатності методу оцінки ефективності при подачі вхідних даних неправильного формату, а також працездатність побудованої моделі. 4.3. Вимоги до умов експлуатації: встановлення мови програмування Python та необхідних бібліотек для роботи методу оцінки ефективності. 4.4. Вимоги до складу і параметрів технічних засобів: комп’ютер або ноутбук із 4 ГБ оперативної пам’яті та процесором не нижче Intel Core i3 9-го покоління.

<table>
<tr><td></td><td colspan="2">4.5. Вимоги до інформаційної та програмної сумісності: ОС Linux або Windows 8/10, підтримка Anaconda, Python 3.
4.6. Вимоги до маркування та упаковки: жорстка упаковка з пластмаси, маркування на українській та англійській мовах.
4.7. Вимоги до транспортування і зберігання: транспортування в упаковці будь-яким способом, температура транспортування/зберігання +5-+20 С.
4.8. Спеціальні вимоги: відсутні.</td></tr>
<tr><td>5. Вимоги до програмної документації.</td><td colspan="2">Програмною документацією до виробу «Метод оцінки ефективності вебсайтів» вважати:
1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).
2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до кваліфікаційної роботи).
3) Опис програмного виробу (представити в розділах 3, 4 пояснювальної записки до кваліфікаційної роботи).
4) Текст програми (представити в Додатку Г до пояснювальної записки до кваліфікаційної роботи).</td></tr>
<tr><td>6. Техніко економічні показники</td><td colspan="2">Оцінити якість побудованих моделей оцінки ефективності вебсайтів за набором показників: MAE, MSE, RMSE, R^2. Виконати у розділі 3 пояснювальної записки до кваліфікаційної роботи.</td></tr>
<tr><td rowspan="2">7. Стадії і етапи розробки</td><td>Дата</td><td>Назва етапу</td></tr>
<tr><td>08.12.2022 – 01.02.2023

01.02.2023 – 15.03.2023
15.03.2023 – 30.04.2023</td><td>1. Аналіз наукової та профільної літератури з підходів та методології оцінки ефективності роботи вебсайтів;
2. Порівняння та аналіз існуючих методів оцінки ефективності вебсайтів;
3. Дослідження можливостей впровадження алгоритмів</td></tr>
</table>

	30.04.2023 – 01.07.2023 01.07.2023 – 30.09.2023 30.09.2023 – 30.10.2023 30.10.2023 – 28.11.2023 12.12.2023	машинного навчання для удосконалення методів оцінки ефективності вебсайтів; 4. Розробка методу оцінки ефективності вебсайтів; 5. Програмна реалізація і тестування методу оцінки ефективності роботи веб-сайтів на реальному наборі даних; 6. Дослідження основних показників якості роботи методу та можливостей їх покращення; 7. Аналіз отриманих результатів та висновки; 8. Представлення кваліфікаційної роботи.
8. Порядок контролю і приймання	Загальні вимоги до приймання розробленого програмного виробу: 1) Перевірка ходу розробки програмного виробу. Керівнику робіт виконувати 1 раз в 3 тижні. 2) Випробування програмного виробу відповідно до Програми і методики випробувань провести на базі комп'ютерного класу або приватного приміщення. 3) Захист розробленого програмного виробу провести на засіданні атестаційної комісії. 4) Пояснювальну записку надати на паперових носіях в одному примірнику, в електронному вигляді.	

Виконавець:
студент групи КУ-61
Чепак А.А.

_____

Замовник:
кандидат технічних наук
Стрілець В.Є.

_____

Додаток В

Програма і методика випробувань програмного виробу

«Метод оцінки ефективності вебсайтів»

1. Об'єкт випробувань

1. Назва програмного виробу: «Метод оцінки ефективності вебсайтів».

2. Галузь застосування: Програмна реалізація методів статистичного аналізу.

3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до дипломної роботи).

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри або приватного приміщення в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного вироби проводяться в обсязі відповідному цієї програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

- працювати на основних операційних системах: Windows, Linux, MacOS;
- вимоги до надійності;
- передбачити захист від некоректних дій користувача;
- сумісність з іншими програмними продуктами;
- бути легко розширюваною;
- елементи програми повинні бути ізольовані одне від одного для зменшення їх впливу на роботи програми під час редагування програмного коду;
- вимоги до складу і параметрів технічних засобів;
- вимоги до маркування та упаковки (не висуваються);
- вимоги до транспортування і зберігання (не висуваються).
- Спеціальні вимоги (не пред’являються).

5. Вимоги до програмної документації

Програмною документацією до виробу «Метод оцінки ефективності веб-сайтів» вважати:

- справжнє технічне завдання на розробку програми (представити як Додаток Б до пояснювальної записки до кваліфікаційної роботи);

– програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи);

– рекомендацій щодо застосування створеної програмної стандартизації у проектах (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи);

– текст програми (представити як Додаток Г до пояснювальної записки до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1. Засоби випробувань

Для проведення випробувань необхідний проект для розробки програмної моделі за допомогою мові програмування Python з використання бібліотек sklearn, numpy, pandas, matplotlib.pyplot, train_test_split, metrics, LinearRegression, Ridge, GridSearchCV, RandomForestRegressor, GradientBoostingRegressor, SVR та вхідним датасетом з характеристиками реальних сайтів.

6.2. Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);

- випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.

2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.

3. Перевірку комплектності складу технічних і програмних засобів.

4. Методику проведення перевірок на 1 етапі випробувань.

5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. Перевірку відповідності технічних характеристик програми вимогам технічного завдання.
2. Перевірку ступеня виконання функціональних вимог до програми.
3. Методику проведення перевірок, що входять до переліку по 2 етапу випробувань.

1. Програма працює відповідно до умов експлуатації операційних систем MS Windows, Linux та MacOS.
2. Для роботи необхідний компілятор мови програмування Python, версії не нижчої ніж 3.0 та файли даних у форматі .csv.
3. Порядок проведення випробувань:

3.1. Перед моделюванням спочатку потрібно завантажити потрібний датасет, використовуючи модуль drive, щоб підключити Google Диск, на якому є потрібний файл з даними. Завантаження здійснюється за допомогою коду Python, в якому потрібно вказати назву датасету:

“pd.read_csv('/content/drive/My Drive/ім’я файлу.csv')”.

3.2. Після чого потрібно запустити програму методу оцінки ефективності.

3.3. Після запускання на екрані з’явиться результат моделювання оцінки ефективності.

Для проведення випробувань пропонується провести тест 1, тест 2, тест 3 та тест 4.

Тест 1

1. Перевірка розміру датасету (кількість рядків там стовпців) та визначення не текстових даних.
2. Запуск програми для аналізу даних.
3. Отримання результатів при успішному виконанні програми.

	Domain	Category	Monthly visit	\
0	google.com	Search_Engines	25.167798	
1	facebook.com	Social_Networks_and_Online_Community	23.558242	
2	twitter.com	Social_Networks_and_Online_Community	22.559521	
3	instagram.com	Social_Networks_and_Online_Community	22.616830	
4	tiktok.com	Social_Networks_and_Online_Community	21.531971	

	Visit duration	Pages	Bounce rate (%)	Social traffic (%)	\
0	10.39	8.69	0.2853	0.0191	
1	10.33	8.70	0.3133	0.0184	
2	10.39	9.88	0.3205	0.0346	
3	8.14	11.01	0.3515	0.0607	
4	3.47	7.58	0.3669	0.0794	

	Display Advertising (%)	Estimation
0	0.005	0.990
1	0.005	0.997
2	0.005	0.995
3	0.005	0.996
4	0.005	0.986

Рисунок В.1 – Тест 1

Тест 2

1. Оцінка навчання та ефективності роботи моделі.
2. Запуск для оцінки навчання та ефективності роботи.
3. Отримання результатів значення оцінки навчання та ефективності моделі.

```
Score on the training set: 1.000
Score on the testing set 0.560
```

Рисунок В.2 – Тест 2

Тест 3

1. Розрахунок метрик регресії (MAE, MSE, RMSE, R^2) для оцінки ефективності моделі та побудова графіку прогнозованого та цільового значення Estimation моделі.
2. Запуск програм для розрахунку метрик та побудови графіку прогнозованого та цільового значення Estimation моделі.
3. Отримання результатів розрахунку метрик та побудови графіку прогнозованого та цільового значення Estimation моделі.

```
Estimation of the Gradient Boosting Model:
_____________________________________
MAE: 0.012029300818788537
MSE: 0.0023101788960496355
RMSE: 0.04806432040557357
R2 Square: 0.9041229330196046
_____________________________________
<ipython-input-35-47f3c8aeddf4>:13: FutureWarning: The frame.append meth
  results_df = results_df.append(results_df_1, ignore_index=True)
```

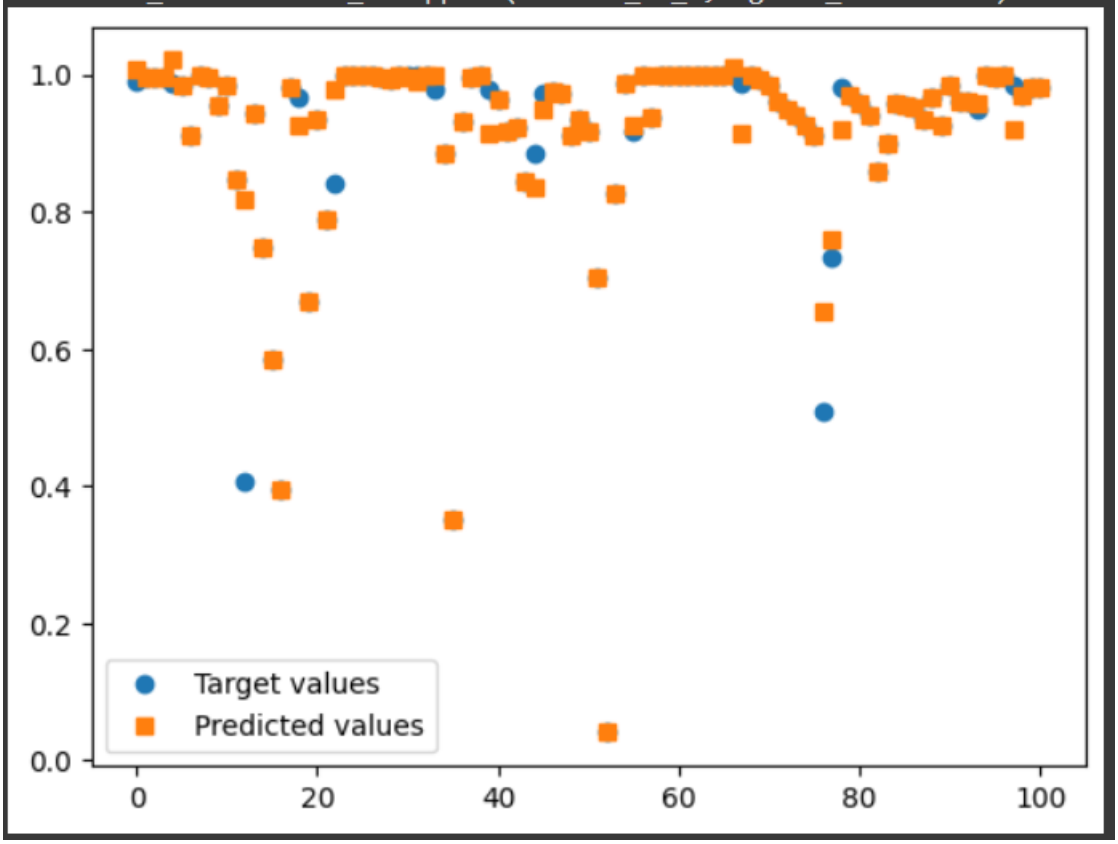

Рисунок В.3 – Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: випробування пройшло успішно, оскільки кожен з тестів показав очікуванні результати.

Виконавець:

студентка групи КУ-61, Чепак А. А.

Додаток Г

Лістинг програмного коду

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

sites = pd.read_csv('/content/drive/My Drive/Sites_data.csv')
print(sites.shape)
print(sites.columns)
print(sites.head())
print(sites.tail())

sites_target= sites['Estimation']
sites_data = sites.drop(['Domain','Category','Estimation'],axis=1)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(sites_data, sites_target, test_size=0.2, random_state=42)

from sklearn import metrics
from sklearn.model_selection import cross_val_score


def print_evaluate(true, predicted):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    print('MAE:', mae)
    print('MSE:', mse)
    print('RMSE:', rmse)
    print('R2 Square:', r2_square)
    print('_____')

def evaluate(true, predicted):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    return mae, mse, rmse, r2_square

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
```

```
lin_model = LinearRegression().fit(X_train, y_train)
ridge_model = Ridge().fit(X_train, y_train)

print("Linear Model")
print("Score on the training set: {:.3f}".format(lin_model.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(lin_model.score(X_test, y_test)))
print("Ridge Model")
print("Score on the training set: {:.3f}".format(ridge_model.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(ridge_model.score(X_test, y_test)))

from sklearn.model_selection import GridSearchCV

ridge_params = {'alpha': [2, 3, 5, 10]}

ridge_grid = GridSearchCV(Ridge(), ridge_params, refit = True, verbose = 1)
ridge_grid.fit(X_train, y_train)

print("Best parameters are ", ridge_grid.best_params_)
print("Best score is ", ridge_grid.best_score_ )

ridge_model_cv = Ridge(alpha=2).fit(X_train, y_train)

print("Score on the training set: {:.3f}".format(ridge_model_cv.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(ridge_model_cv.score(X_test, y_test)))

lin_pred = ridge_model.predict(sites_data)

print('Estimation of the Linear Ridge Model:\n_____')
print_evaluate(sites_target, lin_pred)

plt.figure()
plt.plot(sites_target, 'o', label="Target values")
plt.plot(lin_pred, 's', label="Predicted values")
plt.legend()

results_df = pd.DataFrame(data=[["Linear Ridge Regression", *evaluate(sites_target, lin_pred) ]],
                          columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2 Square'])

from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor

forest_model = RandomForestRegressor().fit(X_train, y_train)
boost_model = GradientBoostingRegressor().fit(X_train, y_train)

print("Random Forest Model")
print("Score on the training set: {:.3f}".format(forest_model.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(forest_model.score(X_test, y_test)))
print("Gradient Boosting Model")
print("Score on the training set: {:.3f}".format(boost_model.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(boost_model.score(X_test, y_test)))
```

```
boost_params = {'max_depth': [3, 4, 5, 7], 'learning_rate':[0.1, 0.3,0.5,1]}

dg_boost_grid = GridSearchCV(GradientBoostingRegressor(), boost_params, refit = True, verbose = 1)
dg_boost_grid.fit(X_train, y_train)

print("Best parameters are ", dg_boost_grid.best_params_)
print("Best score is ", dg_boost_grid.best_score_ )

boost_model_cv = GradientBoostingRegressor(learning_rate= 0.5, max_depth=3).fit(X_train, y_train)

print("Score on the training set: {:.3f}".format(boost_model_cv.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(boost_model_cv.score(X_test, y_test)))

boost_pred = boost_model_cv .predict(sites_data)

print('Estimation of the Gradient Boosting Model:\n_____')
print_evaluate(sites_target, boost_pred)

plt.figure()
plt.plot(sites_target, 'o', label="Target values")
plt.plot(boost_pred, 's', label="Predicted values")
plt.legend()

results_df_1 = pd.DataFrame(data=[["Gradient Boosting Regressor", *evaluate(sites_target, boost_pred)]],
                            columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2 Square'])
results_df = results_df.append(results_df_1, ignore_index=True)

from sklearn.svm import SVR

svm_reg = SVR()
svm_reg.fit(X_train, y_train)

test_pred = svm_reg.predict(X_test)
train_pred = svm_reg.predict(X_train)

print("SVM Model")
print("Score on the training set: {:.3f}".format(svm_reg.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(svm_reg.score(X_test, y_test)))

from sklearn.model_selection import GridSearchCV

svm_params = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}

svm_grid = GridSearchCV(SVR(), svm_params, refit = True, verbose = 1)
svm_grid.fit(X_train, y_train)

print("Best parameters are ", svm_grid.best_params_)
```

```
print("Best score is ", svm_grid.best_score_ )

svm_model_cv = SVR(C= 1000, gamma=0.0001, kernel='rbf').fit(X_train, y_train)

print("Score on the training set: {:.3f}".format(svm_model_cv.score(X_train, y_train)))
print("Score on the testing set {:.3f}".format(svm_model_cv.score(X_test, y_test)))
svm_pred_all = svm_reg.predict(sites_data)

print('Estimation of the SVM Model:\n_____')
print_evaluate(sites_target, svm_pred_all)

plt.figure()
plt.plot(sites_target, 'o', label="Target values")
plt.plot(svm_pred_all, 's', label="Predicted values")
plt.legend()

results_df_2 = pd.DataFrame(data=[["SVM Regressor", *evaluate(sites_target, svm_pred_all)]],
                            columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2 Square'])
results_df = results_df.append(results_df_2, ignore_index=True)

print(results_df)
```