

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАЛЬНО-
НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методика підбору авто з використанням рекомендаційних
алгоритмів»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми Інженерія програмного забезпечення

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Максим ЛЕБІДЬ
(підпис)

Виконав: здобувач вищої освіти група ПДМ-61

_____ Максим ЛЕБІДЬ

Керівник: _____ Наталя ТРІНТІНА
к.т.н., доцент

Рецензент: _____
науковий ступінь, Ім'я, ПРІЗВИЩЕ
вчене звання

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____Ірина ЗАМРІЙ

«_____» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Лебедю Максиму Максимовичу

1. Тема кваліфікаційної роботи: «Методика підбору авто з використанням рекомендаційних алгоритмів»

керівник кваліфікаційної роботи Наталя ТРІНТІНА, к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2024 р..

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методика підбору авто, вимоги до точності рекомендацій

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

1. Аналіз існуючих рекомендаційних систем та алгоритмів.

2. Дослідження методів підбору авто на основі користувацьких вподобань.

3. Розробка методики підбору авто з використанням рекомендаційних алгоритмів.

4. Реалізація системи підбору автомобілів.
5. Тестування та оцінка ефективності розробленої системи.
5. Перелік графічного матеріалу: *презентація*
 1. Мета, об'єкта та предмет дослідження.
 2. Аналіз способів підбору авто.
 3. Аналіз сучасних онлайн-платформ та розробленої методики підбору авто.
 4. Структура рекомендаційного алгоритму.
 5. Інтеграція та функціонування рекомендаційної системи.
 6. Математична модель рекомендаційного алгоритму.
 7. Графічний аналіз експериментальних сценаріїв.
 8. Порівняльна таблиця результатів експериментів.
 9. Архітектура розробленої системи.
 10. Висновки.
 11. Публікації та апробація роботи.
6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10-23.10.2024	
2	Вивчення розвитку рекомендаційних алгоритмів у сфері підбору автомобілів	06.11-12.11.2024	
3	Дослідження методів рекомендаційних систем	13.11-19.11.2024	
4	Аналіз особливостей застосування рекомендаційних алгоритмів для підбору авто	20.11-26.11.2024	
5	Розробка методики підбору авто з використанням рекомендаційних алгоритмів	27.11-03.12.2024	
6	Тестування та оцінка ефективності розробленої методики	04.12-10.12.2023	
7	Оформлення роботи: вступ, висновки, реферат	27.11-03.12.2024	
8	Розробка демонстраційних матеріалів	21.11-24.11.2024	
9	Попередній захист роботи	25.11.2024	

Здобувач вищої освіти

(підпис)

Максим ЛЕБІДЬ

Керівник

кваліфікаційної роботи

(підпис)

Наталя ТРІНТІНА

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 99 стор., 10 табл., 14 рис., 30 джерел.

Мета роботи – оптимізація процесу вибору автомобіля шляхом розробки методики, що зменшує часові показники за рахунок використання рекомендаційних систем.

Об'єкт дослідження – процес підбору автомобілів.

Предмет дослідження – методика підбору автомобілів з використанням рекомендаційних алгоритмів.

У роботі проведено аналіз сучасних рекомендаційних систем та їх застосування в автомобільній індустрії. Розглянуто актуальні проблеми вибору автомобілів в умовах великої різноманітності моделей та комплектацій. Вивчено методи машинного навчання та штучного інтелекту, зокрема алгоритми колаборативної фільтрації, контентного та гібридного підходів, для розробки персоналізованих рекомендацій.

Розроблено методику підбору автомобілів, яка враховує індивідуальні потреби та вподобання користувачів, використовуючи дані про їхні попередні вибори, поведінкові фактори та технічні характеристики автомобілів. Запропоновано алгоритм, що поєднує методи регресійного аналізу та глибоких нейронних мереж для виявлення складних взаємозв'язків між параметрами автомобілів та вподобаннями користувачів.

Реалізовано програмне забезпечення, яке використовує розроблену методику для надання персоналізованих рекомендацій. Система інтегрована з базами даних автомобілів та забезпечує швидку обробку великих обсягів інформації завдяки використанню технологій обробки великих даних.

Проведено експериментальні дослідження для оцінки ефективності розробленої методики.

Результати показали зменшення часу необхідного для вибору автомобіля у порівнянні з часом який витрачається традиційними методами підбору.

Результати дослідження підтверджують доцільність та ефективність використання рекомендаційних алгоритмів у сфері підбору автомобілів. Запропонована методика сприяє підвищенню рівня задоволеності клієнтів та відкриває перспективи для подальшого розвитку персоналізованих сервісів у галузі електронної комерції та автомобільної індустрії.

КЛЮЧОВІ СЛОВА: РЕКОМЕНДАЦІЙНІ АЛГОРИТМИ, ПІДБІР АВТОМОБІЛІВ, МАШИННЕ НАВЧАННЯ, ПЕРСОНАЛІЗАЦІЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ОБРОБКА ВЕЛИКИХ ДАНИХ.

ABSTRACT

The text part of the qualification work for obtaining the master's degree: 99 pages, 10 table, 14 figures, 30 sources.

The purpose of the work is to enhance the efficiency and accuracy of the car selection process by developing a methodology using recommendation algorithms.

The object of research is the process of selecting cars.

The subject of research is the methodology of car selection using recommendation algorithms.

An analysis of modern recommendation systems and their application in the automotive industry has been conducted. Current issues in car selection amidst a wide variety of models and configurations have been considered. Methods of machine learning and artificial intelligence have been studied, including algorithms of collaborative filtering, content-based, and hybrid approaches for developing personalized recommendations.

A methodology for car selection has been developed that takes into account individual user needs and preferences, utilizing data about their previous choices, behavioral factors, and technical characteristics of cars. An algorithm combining methods of regression analysis and deep neural networks has been proposed to identify complex relationships between car parameters and user preferences.

Software implementing the developed methodology to provide personalized recommendations has been realized. The system is integrated with car databases and ensures rapid processing of large volumes of information due to the use of big data processing technologies.

Experimental studies have been conducted to evaluate the effectiveness of the developed methodology. The results showed a increase in recommendation accuracy compared to traditional selection methods, as well as a significant reduction in the time required for the user to make a decision.

The research results confirm the feasibility and effectiveness of using recommendation algorithms in the field of car selection. The proposed methodology contributes to increasing customer satisfaction levels and opens up prospects for further development of personalized services in the e-commerce and automotive industries.

KEY WORDS: RECOMMENDATION ALGORITHMS, CAR SELECTION, MACHINE LEARNING, PERSONALIZATION, ARTIFICIAL INTELLIGENCE, BIG DATA PROCESSING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	12
ВСТУП.....	14
1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ПІДБОРУ АВТОМОБІЛІВ: ВІД ТРАДИЦІЙНИХ ПІДХОДІВ ДО РЕКОМЕНДАЦІЙНИХ АЛГОРИТМІВ.....	16
1.1 Аналіз сучасних методів підбору автомобілів.....	16
1.1.1 Традиційні підходи до підбору автомобілів.....	16
1.1.2 Рекомендаційні алгоритми у підборі автомобілів.....	19
1.1.3 Недоліки та виклики традиційних і сучасних підходів.....	22
1.2 Технології для покращення підбору автомобілів.....	23
1.2.1 Використання машинного навчання.....	24
1.2.2 Інтеграція систем рекомендацій у платформи електронної торгівлі	25
1.2.3 Роль експертних систем і аналітики даних.....	26
1.3 Огляд існуючих рішень у підборі автомобілів.....	28
1.3.1 Сучасні онлайн-платформи та сервіси.....	28
1.3.2 Інструменти для роботи з даними.....	31
1.3.3 Методи взаємодії з користувачами.....	34
1.3.4 Оцінка ефективності існуючих рішень.....	37
2 РОЗРОБКА МЕТОДИКИ ПІДБОРУ АВТОМОБІЛІВ З ВИКОРИСТАННЯМ РЕКОМЕНДАЦІЙНИХ АЛГОРИТМІВ.....	40
2.1 Постановка задачі.....	40
2.2 Основні формули та критерії традиційного методу підбору автомобіля та методу підбору автомобіля на основі штучного інтелекту.....	43
2.3 Збір та обробка великих обсягів даних	46
2.4 Інструменти та технології для роботи з великими даними	47
2.5 Аналіз та вибір алгоритмів з подальшою їх адаптацією	49

2.6	Експериментальні дослідження та оцінка ефективності методики...	52
2.7	Оцінка впливу на користувацький досвід.....	57
2.8	Оцінка ефективності методики та порівняння з існуючими підходами.....	58
3	ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕКОМЕНДАЦІЙ.....	61
3.1	Використані програмні засоби.....	61
3.1.1	Мова програмування та середовище розробки.....	61
3.1.2	Оптимальні інструменти для роботи з даними.....	62
3.1.3	Інтеграція API для рекомендацій.....	64
3.2	Навчання нейронної мережі для рекомендацій.....	65
3.2.1	Підготовка даних для навчання.....	66
3.2.2	Вибір та налаштування архітектури нейронної мережі.....	66
3.2.3	Процес навчання та валідації моделі.....	67
3.3	Реалізація модулів системи.....	67
3.3.1	Опис структури проекту.....	67
3.3.2	Опис інтерфейсу користувача.....	68
3.3.3	Опис розроблених класів та функцій	71
3.3.4	Інтеграція моделі машинного навчання.....	73
3.3.5	Безпека та валідація даних.....	75
3.3.6	Взаємодія з базою даних.....	77
3.4	Аналіз результатів роботи системи.....	79
3.4.1	Тестування рекомендаційного алгоритму.....	79
3.4.2	Аналіз продуктивності.....	81
3.4.3	Висновки щодо роботи системи.....	82
	ВИСНОВКИ.....	83
	ПЕРЕЛІК ПОСИЛАНЬ.....	84
	ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	88
	ДОДАТОК Б. ЛІСТИНГИ ОСНОВНИХ МОДУЛІВ.....	94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ACID (*Atomicity, Consistency, Isolation, Durability*) — атомарність, узгодженість, ізольованість, надійність (властивості транзакцій в базах даних).

AI (*Artificial Intelligence*) — штучний інтелект.

API (*Application Programming Interface*) — інтерфейс програмування додатків.

AWS (*Amazon Web Services*) — хмарні сервіси Amazon.

BI (*Business Intelligence*) — бізнес-аналітика.

CRUD (*Create, Read, Update, Delete*) — створення, читання, оновлення, видалення (базові операції над даними).

CRM (*Customer Relationship Management*) — система управління взаємовідносинами з клієнтами.

CSRF (*Cross-Site Request Forgery*) — міжсайтове підроблення запитів.

ETL (*Extract, Transform, Load*) — витяг, трансформація та завантаження даних.

FAQ (*Frequently Asked Questions*) — часто задавані питання.

HDFS (*Hadoop Distributed File System*) — розподілена файлова система Hadoop.

HTTPS (*HyperText Transfer Protocol Secure*) — захищений протокол передачі гіпертексту.

JSON (*JavaScript Object Notation*) — текстовий формат обміну даними.

KPI (*Key Performance Indicators*) — ключові показники ефективності.

MAE (*Mean Absolute Error*) — середня абсолютна похибка.

ML (*Machine Learning*) — машинне навчання.

MSE (*Mean Squared Error*) — середньоквадратична похибка.

NLP (*Natural Language Processing*) — обробка природної мови.

NoSQL (*Not Only SQL*) — тип баз даних, що не використовують реляційну модель.

ORM (*Object-Relational Mapping*) — об'єктно-реляційне відображення.

PCA (*Principal Component Analysis*) — метод головних компонент.

RDS (*Relational Database Service*) — сервіс реляційних баз даних.

ReLU (*Rectified Linear Unit*) — виправлена лінійна одиниця (функція активації в нейронних мережах).

RMSE (*Root Mean Squared Error*) — корінь середньоквадратичної похибки.

S3 (*Simple Storage Service*) — сервіс простого зберігання даних (від AWS).

SQL (*Structured Query Language*) — мова структурованих запитів.

SSL (*Secure Sockets Layer*) — протокол захисту мережових з'єднань.

t-SNE (*t-distributed Stochastic Neighbor Embedding*) — t-розподілене стохастичне вкладення сусідів.

UI (*User Interface*) — інтерфейс користувача.

UX (*User Experience*) — досвід користувача.

Змінні та позначення в формулах:

a , b — параметри для обчислення часу навчання моделі штучного інтелекту.

β_i — ваговий коефіцієнт для i -го критерію в моделі машинного навчання.

K_i — значення i -го критерію.

K_i норм — нормалізоване значення i -го критерію.

K_i авт — нормалізоване значення i -го критерію для автомобіля.

$K_{\text{макс}}$ — максимальне значення критерію.

$K_{\text{мін}}$ — мінімальне значення критерію.

n — кількість критеріїв.

Num_cars — кількість автомобілів.

Num_params — кількість параметрів.

S — загальний рейтинг автомобіля в традиційному методі.

$T_{\text{навч}}$ — час, необхідний для навчання моделі.

$T_{\text{прогн}}$ — час прогнозування для одного автомобіля.

$T_{\text{традиц}}$ — час виконання традиційним методом.

$T_{\text{ШІ}}$ — час виконання методом штучного інтелекту.

t_0 — час на обробку одного критерію.

W_i — ваговий коефіцієнт для i -го критерію.

Z — прогнозований рейтинг автомобіля в методі штучного інтелекту.

ВСТУП

На сучасному автомобільному ринку спостерігається велика різноманітність моделей і комплектацій, що робить процес вибору автомобіля складним і часозатратним для покупців. Проблема полягає не лише у великій кількості доступних варіантів, але й у тому, що потреби кожного покупця унікальні. Вибір ідеального автомобіля вимагає детального аналізу множини параметрів: від технічних характеристик і вартості до естетичних уподобань та споживчих оцінок.

Ще однією значною складністю є постійне оновлення модельних рядів та технологічних інновацій, які вимагають від покупців постійного бути в курсі новинок і змін на ринку. Це може призводити до інформаційного перевантаження та підвищення стресу під час прийняття рішення, особливо коли покупці не мають значного досвіду в автомобільній сфері.

Системи рекомендацій, які застосовують штучний інтелект та машинне навчання, мають потенціал розв'язання цієї проблеми за допомогою персоналізованих пропозицій, заснованих на широкому спектрі даних, включаючи індивідуальні переваги користувачів, історію покупок та навіть поведінкові фактори. Це дозволяє не тільки оптимізувати процес вибору, але й забезпечити більш високий рівень задоволеності клієнтів, оскільки вони отримують рекомендації, які максимально відповідають їхнім особистим вимогам та очікуванням

Мета роботи полягає у зменшенні часу, необхідного для вибору автомобіля, шляхом розробки методики підбору за рахунок використання рекомендаційних алгоритмів.

Об'єкт дослідження – процес підбору автомобілів.

Предмет дослідження – методика підбору автомобілів з використанням рекомендаційних алгоритмів.

Для досягнення мети здійснювались наступні дії:

1. Літературний огляд. Проведено дослідження та аналіз існуючих

наукових джерел з акцентом на методах та технологіях, що сприяють зменшенню часу вибору товарів за допомогою рекомендаційних систем. Вивчено сучасні алгоритми машинного навчання та штучного інтелекту, які забезпечують швидке генерування персоналізованих рекомендацій.

2. Виявлення факторів, що мають вплив на тривалість вибору. Проаналізовано традиційний процес вибору автомобіля та ідентифіковано ключові фактори, які призводять до збільшення часу, необхідного для прийняття рішення користувачем.

3. Розробка методики підбору автомобілів для зменшення часу вибору. Створено методику, яка використовує рекомендаційні алгоритми, оптимізовані для швидкого оброблення даних та надання рекомендацій у режимі реального часу. Методика враховує індивідуальні вподобання користувачів та технічні характеристики автомобілів.

4. Моделювання та програмування. Розроблено та реалізовано програмне забезпечення, що втілює запропоновану методику. Забезпечено оптимізацію алгоритмів для швидкого виконання та мінімізації часу відгуку системи.

5. Експериментальні дослідження часу вибору. Було проведено серію експериментів для визначення оцінки ефективності розробленої методики з точки зору зменшення часу, необхідного для вибору автомобіля. Порівняно показники часу роботи системи з традиційними методами підбору.

6. Аналіз результатів та оптимізація. Проаналізовано отримані результати, визначено ступінь зменшення часу вибору. На основі аналізу проведено додаткову оптимізацію алгоритмів та системи в цілому для подальшого скорочення часу.

Вирішення цих завдань безпосередньо сприяло досягненню мети зменшення часу, необхідного для вибору автомобіля користувачем, шляхом розробки та впровадження ефективної методики підбору з використанням рекомендаційних алгоритмів.

1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ПІДБОРУ АВТОМОБІЛІВ: ВІД ТРАДИЦІЙНИХ ПІДХОДІВ ДО РЕКОМЕНДАЦІЙНИХ АЛГОРИТМІВ

1.1 Аналіз сучасних методів підбору автомобілів

У сучасному світі вибір автомобіля перетворився на складний процес, який вимагає детального аналізу та врахування багатьох факторів, таких як технічні характеристики, фінансові можливості, особисті потреби та переваги. Традиційні методи підбору, засновані на особистих рекомендаціях або обмеженій інформації, часто не відповідають сучасним вимогам споживачів. З розвитком технологій з'явилися нові підходи, зокрема рекомендаційні алгоритми та системи машинного навчання, які значно спрощують процес вибору. У цьому розділі буде проведено детальний аналіз як традиційних, так і сучасних методів підбору автомобілів, а також розглянуто їхні переваги та недоліки в контексті сучасного ринку[1].

1.1.1 Традиційні підходи до підбору автомобілів

Традиційні методи підбору автомобілів базуються на особистому досвіді, рекомендаціях близьких людей та безпосередньому ознайомленні з автомобілями. Ці підходи були основними до широкого впровадження цифрових технологій і досі залишаються актуальними для багатьох споживачів.

Одним із найпоширеніших способів вибору автомобіля є звернення за порадою до друзів, родичів або колег, що часто називають особистими рекомендаціями або "сарафанним радіо". Люди довіряють думці близьких, які вже мають досвід володіння певною маркою чи моделлю автомобіля. Такий метод дозволяє отримати чесну та неупереджену інформацію про реальні переваги та недоліки автомобіля.

Фізичний огляд автомобілів у автосалонах та дилерських центрах надає можливість ознайомитися з дизайном та інтер'єром автомобіля, отримати консультацію від професійних продавців і провести тест-драйв для оцінки

динаміки та комфорту. Відвідування декількох дилерів дозволяє порівняти різні марки та моделі, а також умови покупки та сервісного обслуговування.

До появи інтернету друковані автомобільні журнали та каталоги були основним джерелом інформації про новинки ринку та технічні характеристики автомобілів. Вони містили детальні огляди та тести автомобілів, порівняльні характеристики різних моделей, рекомендації експертів та рейтинги. Для пошуку вживаних автомобілів традиційно використовувалися газетні оголошення в місцевих виданнях та фізичні дошки оголошень у громадських місцях або спеціалізованих виданнях. Цей метод дозволяв знайти автомобіль у своєму регіоні та безпосередньо зв'язатися з продавцем.

Перед покупкою автомобіля багато хто звертається до незалежних механіків для перевірки технічного стану вживаного автомобіля або до автомобільних експертів для отримання поради щодо вибору моделі відповідно до потреб та бюджету. Це допомагає уникнути прихованих проблем і зробити більш обґрунтований вибір.

Серед переваг традиційних методів можна виділити довіру до джерела інформації, оскільки особисті контакти та рекомендації підвищують рівень довіри. Фізичний контакт з автомобілем дозволяє оцінити його особисто, а індивідуальний підхід забезпечує можливість отримати спеціальні умови або знижки від продавців у автосалонах.

Проте ці методи мають і свої слабкі сторони. Обмежений доступ до даних ускладнює оперативне порівняння великої кількості моделей. Суб'єктивний характер рекомендацій може спричинити упередженість, оскільки вони ґрунтуються на індивідуальних перевагах. До того ж, значно збільшуються витрати часу та ресурсів через необхідність особисто відвідувати численні локації.

Табл. 1.1 наводить порівняння традиційних методів підбору автомобілів, що підкреслює їх основні переваги та недоліки.

Таблиця 1.1

Порівняння традиційних методів підбору автомобілів

Метод	Переваги	Недоліки
Особисті рекомендації	Довіра до джерела, реальний досвід	Суб'єктивність, обмежений вибір
Відвідування автосалонів	Фізичний огляд, професійна консультація, тест-драйв	Витрати часу, можливий тиск з боку продавців
Друковані видання та каталоги	Детальні огляди, експертні оцінки	Неактуальність інформації, відсутність інтерактивності
Оголошення в газетах	Прямий контакт з продавцем, можливість торгу	Ризик шахрайства, обмежена інформація про автомобіль
Консультації з експертами	Професійна оцінка, зниження ризику помилок	Додаткові витрати, залежність від компетентності експерта

Джерела для отримання додаткової інформації

1. Друковані видання:
2. Офіційні дилери та автосалони:
3. Консультації з фахівцями:

Попри розвиток цифрових технологій, традиційні способи підбору автомобілів усе ще залишаються важливими у процесі прийняття рішення. Вони надають можливість особистого ознайомлення з автомобілем та отримання порад від довірених осіб. Проте обмеженість доступної інформації та витрати часу спонукають до пошуку нових, більш ефективних методів, які поєднують традиційний досвід з сучасними технологічними рішеннями.

1.1.2 Рекомендаційні алгоритми у підборі автомобілів

У сучасному цифровому світі рекомендаційні алгоритми стали невід'ємною частиною багатьох сервісів, включаючи підбір автомобілів. Вони дають можливість обробляти значні обсяги даних і пропонують користувачам персональні пропозиції, враховуючи їхні вподобання та поведінку[2].

Сутність рекомендаційних алгоритмів

Рекомендаційні алгоритми — це алгоритми машинного навчання, що застосовуються для прогнозування вподобань користувачів та надання релевантних рекомендацій[3]. У підборі автомобілів вони можуть враховувати:

- Персональну інформацію: вік, стать, місце проживання.
- Поведінкові дані: історія пошуків, переглянутих та обраних автомобілів.
- Відгуки та оцінки: рейтинги, залишені користувачами.
- Тренди ринку: популярність моделей, сезонні зміни попиту.

Типи рекомендаційних систем

1. Контентно-орієнтовані системи[4] (Content-Based Filtering)

Аналізують характеристики автомобілів (марка, модель, тип двигуна) та вподобання користувача для надання рекомендацій. Якщо користувач цікавиться певною маркою або типом кузова, система пропонує схожі варіанти.

2. Колаборативна фільтрація[5] (Collaborative Filtering)

Базується на аналізі поведінки інших користувачів з подібними вподобаннями. Якщо інші користувачі з аналогічним профілем обрали певні автомобілі, вони будуть рекомендовані і поточному користувачу.

3. Гібридні системи

Поєднують обидва підходи для підвищення точності рекомендацій. Вони враховують як характеристики автомобілів, так і взаємодію користувачів з системою.

Рекомендаційні алгоритми широко застосовуються у підборі автомобілів, особливо в сучасних цифрових платформах. Онлайн-платформи з продажу автомобілів, такі як AutoTrader та Cars.com, активно використовують ці

алгоритми для персоналізації контенту. Вони аналізують дії користувачів, їхні пошукові запити та поведінку на сайті, щоб пропонувати релевантні оголошення та рекомендації, які максимально відповідають інтересам кожного користувача.

Мобільні додатки, наприклад Carvana, також впроваджують рекомендаційні системи для покращення користувацького досвіду. Завдяки аналізу попередніх виборів та вподобань користувача, додатки пропонують автомобілі, які можуть його зацікавити, тим самим спрощуючи процес пошуку та вибору.

Автодилери використовують рекомендаційні алгоритми в своїх CRM-системах для ефективнішого управління взаєминами з клієнтами. Це дозволяє їм пропонувати спеціальні пропозиції та акції, які відповідають потребам та інтересам конкретного клієнта, підвищуючи рівень задоволеності та лояльності.

Переваги використання рекомендаційних алгоритмів у підборі автомобілів очевидні. По-перше, це персоналізація: користувач отримує пропозиції, що відповідають його індивідуальним вподобанням та вимогам, що робить процес вибору більш зручним та ефективним. По-друге, це підвищення конверсії: релевантні рекомендації збільшують ймовірність того, що користувач здійснить покупку, оскільки він бачить пропозиції, які дійсно відповідають його потребам. По-третє, це ефективність: алгоритми автоматизують процес підбору, що економить час як для користувача, так і для продавця, і дозволяє обробляти великі обсяги даних швидко та точно.

Проте впровадження рекомендаційних алгоритмів супроводжується певними недоліками та викликами. Однією з ключових проблем є забезпечення приватності даних. Застосування персональної інформації користувачів для створення рекомендацій викликає питання конфіденційності та безпеки, що може стати причиною занепокоєння серед споживачів і потребує дотримання строгих регуляторних вимог. Іншим викликом є проблема "холодного старту", коли нові користувачі або продукти не мають достатньо історичних даних, що призводить до менш точних рекомендацій і може знижувати ефективність системи на початкових етапах. Крім того, складність реалізації таких алгоритмів потребує значних ресурсів та експертних знань. Розробка та впровадження ефективних

рекомендаційних систем вимагає висококваліфікованих спеціалістів, часу та фінансових інвестицій, що може бути недоступним для деяких компаній або проектів.

Таким чином, рекомендаційні алгоритми відіграють важливу роль у сучасному підборі автомобілів, пропонуючи значні переваги для користувачів так для продавців. Вони сприяють персоналізації, підвищенню конверсії та ефективності процесів. Однак, для успішного впровадження необхідно враховувати питання приватності, вирішувати проблему "холодного старту" та бути готовим до інвестицій у складні технологічні рішення(Табл. 1.2).

Таблиця 1.2

Порівняння типів рекомендаційних систем

Тип системи	Переваги	Недоліки
Контентно-орієнтована	Враховує індивідуальні вподобання, незалежна від інших користувачів	Може не враховувати популярність або тренди
Колаборативна фільтрація	Виявляє приховані зв'язки між користувачами та продуктами	Чутлива до проблеми "холодного старту", потребує багато даних
Гібридна	Поеднує переваги обох підходів, підвищує точність	Складна в реалізації, вимагає більше ресурсів

Реалізацію рекомендаційних алгоритмів у підборі автомобілів можна проілюструвати на Рисунку 1.1, який показує мапу етапів рекомендаційного алгоритму. Наприклад, сервіс Amazon Personalize є хмарним рішенням, що дозволяє інтегрувати рекомендаційні алгоритми в додатки та сайти, включаючи платформи з продажу автомобілів[6]. Інший приклад — AutoScout24, європейська платформа, яка використовує алгоритми для рекомендації автомобілів на основі поведінки користувача.

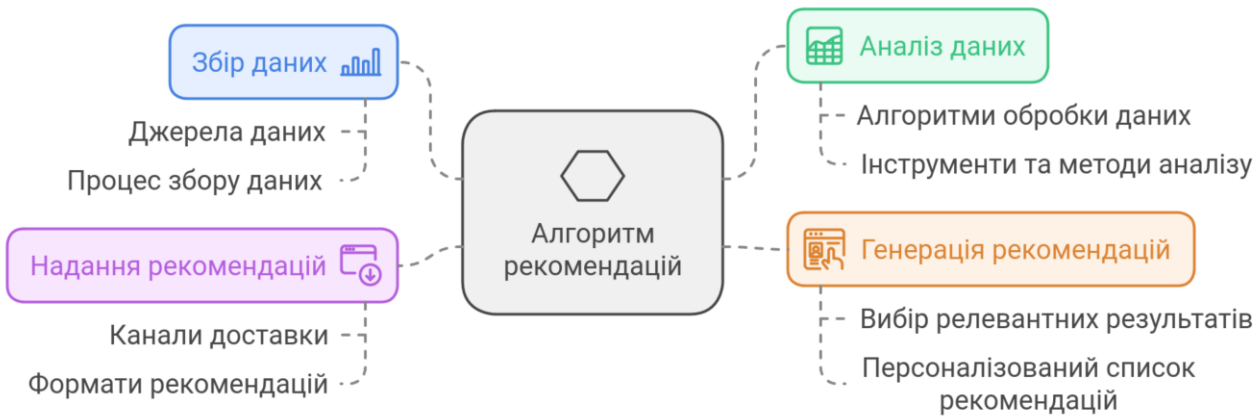


Рис. 1.1 Мапа етапів рекомендаційного алгоритму

1.1.3 Недоліки та виклики традиційних і сучасних підходів

Традиційні та сучасні методи мають свої переваги, але також стикаються з певними недоліками та викликами, які можуть вплинути на ефективність процесу вибору.

Недоліки традиційних підходів (Табл. 1.3)

Таблиця 1.3

Порівняльна таблиця недоліків та викликів

Аспект	Традиційні підходи	Сучасні підходи
Доступ до інформації	Обмежений, може бути застарілим	Широкий, але ризик перевантаження інформацією
Об'єктивність рекомендацій	Суб'єктивні, базуються на особистих вподобаннях	Залежить від алгоритму та якості даних
Часові та фінансові витрати	Високі через необхідність фізичного відвідування та консультацій	Нижчі, процес автоматизований
Приватність та безпека даних	Менш актуальні, оскільки мало персональних даних збирається	Підвищений ризик порушення конфіденційності та компрометації даних.
Технічні вимоги	Низькі, не потребують спеціальних технологій	Високі, потребують стабільного інтернет-з'єднання та сучасних пристроїв

Продовження таблиці 1.3

Порівняльна таблиця недоліків та викликів

Аспект	Традиційні підходи	Сучасні підходи
Точність рекомендацій	Може бути низькою через обмежену інформацію та суб'єктивність	Висока при наявності якісних даних, але може постраждати від проблем з даними
Адаптивність	Низька, складно швидко реагувати на зміни ринку та вподобань	Висока, алгоритми можуть швидко адаптуватися до нових трендів
Ризик шахрайства	Можливий через недобросовісних продавців у приватних оголошеннях	Зменшується завдяки системам перевірки та відгукам користувачів

Виклики для обох підходів

- Поєднання підходів

Знайти баланс між традиційними та сучасними методами для максимального задоволення потреб користувача.

- Освіта та довіра користувачів

Підвищення рівня довіри до нових технологій та навчання користувачів ефективному використанню сучасних інструментів.

1.2 Технології для покращення підбору автомобілів

Сучасні технології суттєво трансформують процес підбору автомобілів, надаючи споживачам та бізнесу інструменти для більш ефективного та персоналізованого вибору. Застосування машинного навчання, впровадження рекомендаційних систем у платформи електронної торгівлі, а також використання експертних систем і аналітики даних сприяють оптимізації процесів, підвищенню рівня задоволеності клієнтів та зростанню обсягів продажів.

1.2.1 Використання машинного навчання

Машинне навчання (ML) — це напрямок штучного інтелекту, що дозволяє системам самостійно навчатися й удосконалюватися, спираючись на накопичений досвід без необхідності прямого програмування. У контексті вибору автомобілів машинне навчання застосовується для аналізу великих масивів даних і виявлення шаблонів, що сприяють прийняттю обґрунтованих рішень [7].

Одним із застосувань машинного навчання у підборі автомобілів є прогнозування цін. Алгоритми машинного навчання аналізують історичні дані про продажі автомобілів, враховуючи параметри, такі як марка, модель, рік випуску, пробіг та стан транспортного засобу. За допомогою цього, можна точно оцінити ринкову вартість автомобіля та прогнозувати її зміни.

Крім того, машинне навчання використовується для аналізу відгуків та рейтингів. Застосування методів обробки природної мови (NLP) системи обробляють текстові відгуки, щоб виявити позитивні та негативні аспекти конкретних моделей автомобілів.

Серед методів машинного навчання, що використані у цій сфері, можна виділити регресію для прогнозування числових значень, таких як ціна або пробіг автомобіля; класифікацію для визначення категорії автомобіля, наприклад, економ-клас чи преміум; а також кластеризацію. Використання цих алгоритмів дає змогу значно покращити точність і ефективність рекомендацій, що ілюструється у Таблиці 1.4.

Таблиця 1.4

Приклади застосування машинного навчання у підборі автомобілів

Задача	Алгоритм	Результат
Прогнозування ціни	Лінійна регресія	Оцінка вартості автомобіля на основі характеристик
Персоналізовані рекомендації	Колаборативна фільтрація	Список автомобілів, що можуть зацікавити користувача

Приклади застосування машинного навчання у підборі автомобілів

Задача	Алгоритм	Результат
Аналіз відгуків	Обробка природної мови (NLP)	Виявлення основних тем та настроїв у відгуках
Виявлення шахрайства	Алгоритми виявлення аномалій	Ідентифікація підозрілих оголошень

1.2.2 Інтеграція систем рекомендацій у платформи електронної торгівлі

Рекомендаційні системи стали значущою частиною платформ електронної торгівлі, надаючи користувачам персоналізований досвід та сприяючи збільшенню продажів. Ключовими аспектами інтеграції є персоналізація контенту, покращення навігації та збільшення залученості користувачів. Персоналізація контенту передбачає відображення автомобілів та пропозицій, які відповідають інтересам користувача, використовуючи історію пошуку та поведінку для формування релевантних рекомендацій. Покращення навігації досягається завдяки системам, що допомагають користувачам швидше знаходити потрібні моделі за допомогою фільтрів та сортування. Збільшення залученості забезпечується інтерактивними елементами, такими як рекомендації "Схожі автомобілі" або "Популярні моделі", які утримують увагу користувачів на платформі.

Як показано на рисунку 1.2, інтеграційний цикл рекомендованої системи забезпечує послідовність етапів, що сприяють персоналізації користувацького досвіду. Цикл включає аналіз поведінкових даних, обробку отриманої інформації алгоритмами рекомендацій та відображення персоналізованих результатів для користувача. Ця структура дозволяє оптимізувати взаємодію між платформою і користувачем, підвищуючи ефективність системи.

Переваги такої інтеграції очевидні як для бізнесу, так і для користувачів. Для бізнесу це означає підвищення конверсії та середнього чека, а також кращу сегментацію аудиторії для маркетингових кампаній. Для користувачів інтеграція

рекомендаційних систем забезпечує економію часу при пошуку та надає більш релевантні й цікаві пропозиції.



Рис. 1.2 Інтеграційний цикл рекомендованої системи

1.2.3 Роль експертних систем і аналітики даних

Експертні системи та аналітика даних є ключовими інструментами для прийняття обґрунтованих рішень і оптимізації бізнес-процесів у сфері підбору автомобілів. Вони дозволяють автоматизувати складні процеси, базуючись на правилах експертів і аналізі даних.

Експертні системи є програмними рішеннями, які використовують знання та правила, закладені експертами, для вирішення завдань. У сфері підбору автомобілів ці системи застосовуються для надання консультацій, які враховують індивідуальні потреби користувачів та експертні знання про автомобілі. Крім того, експертні системи здатні проводити діагностику стану автомобіля та виявляти можливі проблеми на основі введених параметрів.

Аналітика даних, зі свого боку, сприяє глибшому розумінню ринку та клієнтів[8]. Вона включає аналіз значних обсягів даних для пошуку тенденцій у попиті, цінах і вподобаннях споживачів. Важливим компонентом аналітики є

сегментація клієнтів, яка дозволяє розподілити клієнтів на групи за критеріями, такими як вік, бюджет або інтереси, що робить маркетингові кампанії більш цільовими. Крім того, завдяки статистичним методам і алгоритмам машинного навчання аналітика даних використовується для прогнозування майбутніх тенденцій і попиту.

Як показано у таблиці 1.5, експертні системи та аналітика даних мають різні підходи та завдання, що підкреслює їхню взаємодоповнюваність. Експертні системи базуються на правилах і знаннях, отриманих від фахівців, і забезпечують прийняття рішень та консультації. Водночас аналітика даних застосовує статистичні методи і машинне навчання для виявлення закономірностей і прогнозування. Наприклад, експертні системи можуть рекомендувати моделі автомобілів, тоді як аналітика даних прогнозує ринкові тенденції, допомагаючи приймати стратегічні рішення[9].

Переваги інтеграції експертних систем і аналітики даних включають автоматизацію процесів, що знижує людський фактор, покращення якості рішень завдяки інформаційній об'єктивності, а також оптимізацію ресурсів через ефективне використання бюджету і часу.

Таблиця 1.5

Порівняння експертних систем та аналітики даних

Критерій	Експертні системи	Аналітика даних
Підхід	Базується на правилах та знаннях експертів	Застосовує статистичні методи та ML
Джерело знань	Експертні знання	Дані (структуровані та неструктуровані)
Основна задача	Прийняття рішень, консультації	Виявлення закономірностей, прогнозування
Приклад застосування	Рекомендації з вибору моделі	Прогнозування ринкових тенденцій

1.3 Огляд існуючих рішень у підборі автомобілів

На ринку з'явилися різноманітні онлайн-платформи та сервіси, які пропонують користувачам швидкий та зручний доступ до широкого асортименту автомобілів. Крім того, розроблено інструменти для роботи з великими обсягами даних, що дозволяє більш точно відповідати потребам споживачів. У цьому розділі буде проведено детальний аналіз наявних рішень у підборі автомобілів, включаючи сучасні онлайн-платформи, інструменти для обробки даних, методи співпраці з користувачами та оцінку ефективності цих рішень. Це дозволить визначити найкращі практики та окреслити напрямки для подальшого розвитку галузі.

1.3.1 Сучасні онлайн-платформи та сервіси

У цифрову епоху онлайн-платформи стали основним інструментом для підбору та покупки автомобілів. Вони надають користувачам можливість швидкого доступу до широкого асортименту транспортних засобів, фільтрації за різними параметрами та отримання детальної інформації про кожну модель. У цьому розділі розглянемо одну глобальну платформу та декілька українських сервісів, які є лідерами на ринку онлайн-продажу автомобілів.

Глобальна платформа: AutoScout24

AutoScout24 є однією з найбільших європейських онлайн-платформ для продажу нових та вживаних автомобілів. Заснована в Німеччині, вона охоплює більшість країн Європи, надаючи доступ до мільйонів оголошень. Серед особливостей AutoScout24 можна виділити широкий асортимент — більше 2 мільйонів оголошень з різних країн Європи. Платформа пропонує зручні фільтри пошуку, що дозволяють користувачам відбирати автомобілі за маркою, моделлю, ціною, пробігом, типом палива, коробкою передач та іншими параметрами. Використання рекомендаційних систем дозволяє пропонувати автомобілі на основі попередніх пошуків та вподобань користувача. Крім того, AutoScout24 має мобільний додаток для iOS та Android, що дає змогу шукати автомобілі на ходу.

Безпека та надійність забезпечуються через верифікацію продавців та систему оцінок, що допомагає уникнути шахрайства.

Українські онлайн-платформи

Однією з провідних українських онлайн-платформ є AUTO.RIA.com, яка є найбільшою в Україні для продажу нових та вживаних автомобілів. Платформа має велику базу оголошень — понад 200 000 актуальних пропозицій з усієї країни. AUTO.RIA.com пропонує перевірені автомобілі з детальними звітами про стан та історію, що підвищує довіру користувачів. Інтуїтивно зрозумілий дизайн та зручні фільтри пошуку покращують користувацький досвід. Також платформа має мобільний додаток для iOS та Android, а рекомендаційні алгоритми пропонують автомобілі на основі вподобань та історії пошуків користувача.

Іншою популярною платформою є OLX.ua, яка має розділ "Транспорт". Це популярна в Україні платформа оголошень, де можна знайти різноманітні пропозиції — від автомобілів до мотоциклів, вантажівок та іншого транспорту. Простота розміщення оголошень дозволяє приватним особам легко додавати свої пропозиції. OLX.ua має систему повідомлень, що дає можливість спілкуватися з продавцями безпосередньо через платформу. Мобільний додаток доступний для різних мобільних платформ, що робить сервіс зручним для широкого кола користувачів.

Спеціалізованим автомобільним порталом України є RST.ua. Сервіс фокусується виключно на продажу та купівлі автомобілів. Оновлення в реальному часі забезпечує появу нових оголошень щохвилини, що дозволяє користувачам оперативно знаходити актуальні пропозиції. Платформа надає детальні оголошення з великою кількістю фотографій та детальним описом кожного автомобіля. Крім того, RST.ua має розділ новин з останніми подіями автомобільного ринку та оглядами новинок.

Таблиця 1.4 наводить порівняльний аналіз цих платформ, що допомагає чітко побачити їх основні характеристики та відмінності.

Таблиця 1.6

Порівняльний аналіз платформ

Критерій	AutoScout24	AUTO.RIA.com	OLX.ua	RST.ua
Географічне охоплення	Європа	Україна	Україна	Україна
Кількість оголошень	Більше 2 млн	Понад 200 000	Понад 100 000 в розділі "Транспорт"	Понад 150 000
Спеціалізація	Автомобілі (нові та вживані)	Автомобілі (нові та вживані)	Різні категорії товарів, включаючи транспорт	Автомобілі
Мобільний додаток	iOS, Android	iOS, Android	iOS, Android	Немає офіційного додатку
Рекомендаційні системи	Є	Є	Обмежено	Немає
Критерій	AutoScout24	AUTO.RIA.com	OLX.ua	RST.ua

Методи взаємодії з користувачами

Онлайн-платформи активно використовують різноманітні методи для залучення та утримання користувачів. Персоналізація контенту на основі історії пошуків та переглядів дозволяє пропонувати релевантні оголошення, що підвищує інтерес користувачів. Сповіщення про нові оголошення, зниження цін або відповіді продавців допомагають користувачам залишатися в курсі актуальних пропозицій. Інтеграція з соціальними мережами спрощує процес входу на платформу та дозволяє ділитися оголошеннями з друзями. Служба підтримки, включаючи онлайн-чати та гарячі лінії, забезпечує оперативне вирішення питань та підвищує рівень задоволеності користувачів.

Технологічні особливості

Сучасні платформи забезпечують адаптивний дизайн, оптимізований для різних пристроїв — комп'ютерів, планшетів, смартфонів, що робить доступ до сервісу зручним у будь-який час. Безпека даних користувачів гарантується за допомогою SSL-сертифікатів та інших заходів захисту персональної інформації. Інтеграція з картографічними сервісами дозволяє відображати місцезнаходження автомобіля на карті, що полегшує планування поїздки до продавця та оцінку логістичних аспектів покупки.

Виклики та перспективи

Ринок онлайн-продажу автомобілів є висококонкурентним, тому платформи постійно впроваджують нові функції та сервіси для утримання та розширення аудиторії. Питання безпеки, зокрема ризик шахрайства та недостовірних оголошень, залишаються актуальними і вимагають від платформ постійного вдосконалення систем модерації та верифікації користувачів. Технологічний розвиток відкриває нові перспективи: впровадження штучного інтелекту, віртуальної та доповненої реальності може суттєво покращити користувацький досвід, наприклад, через віртуальні тури автомобілями або більш точні рекомендаційні системи.

Сучасні онлайн-платформи та сервіси значно спростили процес підбору автомобілів для споживачів, надаючи доступ до широкого асортименту пропозицій та засобів для їх порівняння. Глобальні платформи, такі як AutoScout24, охоплюють міжнародний ринок, тоді як українські сервіси, такі як AUTO.RIA.com та OLX.ua, адаптовані до місцевих особливостей та потреб споживачів. Постійний розвиток технологій та впровадження інновацій забезпечують конкурентоспроможність цих платформ і підвищують задоволеність користувачів.

1.3.2 Інструменти для роботи з даними

У сучасному підборі автомобілів ефективна робота з даними є критично важливою. Інструменти для інформаційної обробки допомагають збирати,

зберігати, аналізувати та візуалізувати інформацію, що дозволяє приймати обґрунтовані рішення, розробляти персоналізовані рекомендації та підвищувати задоволеність клієнтів.

Однією з основних категорій є системи керування базами даних (СУБД). Реляційні СУБД, такі як MySQL, PostgreSQL та Oracle, використовуються для зберігання структурованих даних, наприклад, інформації про автомобілі, клієнтів та транзакції. Водночас NoSQL СУБД, як-от MongoDB та Cassandra, підходять для зберігання неструктурованих даних, таких як логи активності користувачів або дані соціальних мереж[10].

Інструменти для збору та обробки даних є ключовими у інтеграції інформації з різноманітних джерел. ETL-платформи, такі як Talend і Apache Nifi, забезпечують автоматизацію процесів збору, трансформації та завантаження даних у сховища. Для вилучення даних з веб-сайтів широко застосовуються інструменти веб-скрейпінгу, зокрема BeautifulSoup і Scrapy для мови Python.

Для аналізу та моделювання даних застосовуються аналітичні та статистичні інструменти. Мови програмування, такі як Python з бібліотеками Pandas і NumPy, та R, широко використовуються для статистичного аналізу. BI-платформи (Business Intelligence), зокрема Tableau, Power BI та QlikView, допомагають у візуалізації даних та створенні інтерактивних звітів, що сприяє кращому розумінню інформації та прийняттю обґрунтованих рішень.

Інструменти машинного навчання та штучного інтелекту є ключовими для побудови моделей прогнозування та класифікації. Бібліотеки та фреймворки, такі як Scikit-learn, TensorFlow та Keras, використовуються для розробки та тренування алгоритмів машинного навчання. Також наявні спеціалізовані платформи для створення моделей, такі як H2O.ai та Google Cloud AI Platform, які спрощують процес впровадження штучного інтелекту в бізнес-процеси.

Хмарні сервіси та платформи відкривають широкі можливості для зберігання й обробки даних. Наприклад, Amazon Web Services (AWS) пропонує рішення, такі як S3 для зберігання даних, RDS для роботи з базами даних, Redshift для аналітики та SageMaker для реалізації проєктів машинного навчання. Схожі

інструменти пропонують і інші провайдери, зокрема Microsoft Azure і Google Cloud Platform, які забезпечують засоби для обробки даних та ML у хмарному середовищі.

У Таблиці 1.7 представлено порівняння основних технологій для роботи з інформацією, таких як Hadoop MapReduce, Apache Spark та NoSQL бази даних[17]. Вони оцінюються за критеріями швидкості, застосування, масштабування та використання, що допомагає визначити найбільш підходящі інструменти для конкретних завдань.

Таблиця 1.7

Порівняння основних технологій для роботи з інформацією

Критерій	Hadoop MapReduce	Apache Spark	NoSQL бази даних
Швидкість	Повільніше, обчислення виконуються на дисках	Швидше, можливість виконання в пам'яті	Залежить від типу бази даних; швидкий доступ до неструктурованих даних
Застосування	Великі пакетні завдання, стійке зберігання	Машинне навчання, потокові обчислення	Широкий спектр застосувань, включно з веб-додатками, кешуванням та великими даними
Масштабування	Лінійне, кластери	Лінійне, можливість виконання у кластерах та хмарах	Горизонтальне масштабування, легко адаптується до великих обсягів даних
Використання	Простіше API для пакетних процесів	Складніше API, але більш гнучке для різних видів завдань	Гнучкі схеми, проста інтеграція з різними додатками

Застосування цих інструментів у підборі автомобілів охоплює декілька ключових напрямків. По-перше, створення персоналізованих рекомендацій шляхом використання ML-моделей для аналізу вподобань користувачів та надання їм релевантних пропозицій. По-друге, аналіз ринкових тенденцій дозволяє виявляти тренди у попиті на певні моделі або характеристики автомобілів, що допомагає коригувати асортимент та маркетингові стратегії. Також важливим є ціноутворення, де статистичні моделі використовуються для прогнозування оптимальної ціни автомобіля на основі його характеристик та ринкових умов. Нарешті, виявлення шахрайства здійснюється за допомогою алгоритмів, які аналізують аномалії у даних оголошень та активності користувачів з метою запобігання недобросовісним практикам.

1.3.3 Методи взаємодії з користувачами

Дієва співпраця з користувачами є ключовим фактором успіху онлайн-платформ для підбору автомобілів. Методи взаємодії безпосередньо впливають на залученість користувачів, їхню задоволеність сервісом та, як результат, на конверсію і лояльність клієнтів. У цьому розділі розглянуто основні методи взаємодії з користувачами, їхні особливості, переваги та виклики.

Перш за все, інтуїтивний користувацький інтерфейс (UI) та досвід користувача (UX) є фундаментальними складовими успішної взаємодії. Простий та зрозумілий дизайн надає зручну навігацію та швидкий доступ до потрібної інформації, що сприяє позитивному досвіду користувача. Адаптивний дизайн, оптимізований для різних пристроїв—комп'ютерів, планшетів, смартфонів—гарантує зручність використання платформи незалежно від способу доступу. Швидкість завантаження сторінок має прямий вплив на рівень задоволеності користувачів та їхню схильність залишатися на веб-сайті.

Персоналізація та рекомендації є наступним важливим методом взаємодії. Використання рекомендаційних систем, які на основі поведінки користувача пропонують релевантні автомобілі, підвищує релевантність контенту та задовольняє індивідуальні потреби користувача. Налаштування профілю дозволяє

користувачам вказувати свої вподобання, отримуючи персоналізований контент, а динамічний контент, що відображає спеціальні пропозиції та акції, відповідає їхнім інтересам.

Комунікаційні канали відіграють вагомую роль у підтримці зв'язку з користувачами. Онлайн-чати та месенджери забезпечують швидкий зв'язок з продавцями або службою підтримки, дозволяючи оперативно отримувати інформацію. Електронна пошта та SMS-повідомлення використовуються для сповіщень про нові оголошення, зниження цін або відповіді на запити. Push-сповіщення на мобільних пристроях надають миттєвий доступ до актуальних пропозицій.

Інтеграція з соціальними мережами також є ефективним методом взаємодії. Авторизація через соціальні мережі спрощує процес реєстрації та входу на платформу. Можливість ділитися цікавими пропозиціями в соціальних мережах розширює аудиторію та сприяє соціальному доказу, оскільки відгуки та оцінки від інших користувачів підвищують довіру.

Інтерактивні елементи та мультимедіа збагачують користувацький досвід. Детальні фотогалереї та огляди надають візуальну інформацію про автомобіль, а відеоогляди, такі як презентації та тест-драйви, допомагають користувачам краще ознайомитися з моделлю.

Важливою складовою є система відгуків та рейтингів. Можливість залишати оцінки продавців та коментарі до оголошень забезпечує зворотний зв'язок і підвищує довіру. Це допомагає іншим користувачам приймати обґрунтовані рішення, а також стимулює продавців покращувати свій сервіс.

Програми лояльності та гейміфікація спрямовані на утримання клієнтів та стимулювання їхньої активності. Нагородження активних користувачів бонусами та знижками, введення рівнів та статусів, а також проведення конкурсів та акцій залучають аудиторію через інтерактивні заходи.

Нарешті, підтримка та допомога є критично важливими для підвищення задоволеності клієнтів. Наявність розділів FAQ та баз знань, онлайн-підтримка через чат або телефон, а також навчальні матеріали у вигляді статей та відео

допомагають користувачам ефективно використовувати платформу та швидко вирішувати проблеми.

У Таблиці 1.8 наведено порівняння основних методів взаємодії з користувачами, їхні переваги та виклики, що допомагає зрозуміти, на які аспекти слід звернути особливу увагу при розробці та вдосконаленні платформи.

Таблиця 1.8

Порівняння методів взаємодії з користувачами

Метод взаємодії	Переваги	Виклики
Інтуїтивний UI/UX	Покращує користувацький досвід, підвищує залученість	Необхідність постійного оновлення та адаптації
Персоналізація та рекомендації	Підвищує релевантність контенту, збільшує конверсію	Вимоги до захисту даних, складність реалізації
Комунікаційні канали	Швидкий зв'язок, підвищення довіри	Підтримка 24/7, ресурси на обробку запитів
Соціальні мережі	Розширення аудиторії, соціальний доказ	Контроль репутації, інтеграція з різними платформами
Інтерактивні елементи	Більш детальне ознайомлення з продуктом	Вимоги до технологій та ресурсів
Система відгуків та рейтингів	Підвищення довіри, зворотній зв'язок	Модерація контенту, можливі негативні відгуки
Програми лояльності	Утримання клієнтів, стимулювання активності	Розробка ефективної програми, витрати на бонуси
Підтримка та допомога	Підвищення задоволеності клієнтів	Ресурси на навчання персоналу, оновлення матеріалів

Серед викликів при взаємодії з користувачами варто зазначити забезпечення безпеки та конфіденційності. Необхідність захисту персональних даних користувачів відповідно до законодавства є критичною, особливо в контексті зростання кіберзагроз. Підтримка багатомовності є важливою для забезпечення доступності платформи для користувачів з різних регіонів та мовних груп. Адаптація до різних платформ потребує оптимізації сервісу для веб, мобільних додатків та інших гаджетів, що вимагає додаткових ресурсів.

Важливим аспектом є робота з відгуками та негативними коментарями. Необхідно ефективно реагувати на критику та вирішувати проблеми користувачів, що сприяє покращенню репутації та довіри до платформи.

Сучасні тренди у методах взаємодії включають використання чат-ботів та штучного інтелекту. Автоматизація відповідей на запити користувачів за допомогою персональних асистентів не тільки підвищує оперативність, але й зменшує навантаження на службу підтримки.

1.3.4 Оцінка ефективності існуючих рішень

Оцінка ефективності існуючих рішень у підборі автомобілів є важливим етапом для визначення їхнього впливу на ринок, рівня задоволеності клієнтів та виявлення можливостей для вдосконалення. У цьому розділі ми проаналізуємо основні критерії, за якими можна оцінити ефективність сучасних онлайн-платформ та сервісів, і порівняємо їхні показники.

Першим критерієм є користувацький досвід (UX). Він охоплює зручність навігації та пошуку, швидкість завантаження сторінок та адаптивність дизайну під різні пристрої. Якісний UX надає користувачам швидкий доступ до потрібної інформації що сприяє позитивному сприйняттю платформи. Зручна навігація та інтуїтивний інтерфейс дозволяють швидко знаходити потрібні автомобілі, а швидкість завантаження сторінок впливає на задоволеність користувачів і їх бажання залишатися на сайті. Адаптивний дизайн гарантує комфортне користування платформою на комп'ютерах, планшетах та смартфонах.

Другим критерієм є функціональність платформи. Це включає наявність розширених фільтрів та параметрів пошуку, інтеграцію рекомендаційних систем та додаткових сервісів, таких як перевірка VIN-коду, страхування та фінансування. Розширені фільтри допомагають користувачам швидко налаштувати пошук відповідно до їхніх потреб, а рекомендаційні системи пропонують релевантні варіанти на основі їхньої поведінки та вподобань. Додаткові сервіси зроблять процес вибору та придбання автомобіля більш комплексним і зручним.

Третій критерій стосується залученості користувачів. Він оцінюється за такими показниками, як час перебування на сайті, кількість переглянутих сторінок та показники повернення користувачів (*retention rate*). Висока залученість доводить те, що платформа успішно утримує увагу користувачів і задовольняє їхні потреби. Чим довше користувачі залишаються на сайті та чим більше сторінок вони переглядають, тим вищою є ймовірність, що вони знайдуть потрібний автомобіль і здійснять цільову дію.

Конверсія є четвертим критерієм ефективності. Вона визначається відсотком відвідувачів, які здійснили цільову дію, наприклад, зв'язалися з продавцем або залишили заявку, а також кількістю успішних угод через платформу. Високі показники конверсії свідчать про ефективність платформи у перетворенні відвідувачів на реальних клієнтів та забезпеченні успішних транзакцій.

П'ятим критерієм є безпека та довіра. Сюди входить захист персональних даних, наявність верифікації продавців та оголошень, а також система відгуків та рейтингів. Забезпечення конфіденційності інформації користувачів та захист від шахрайства є критично важливими для підтримання високого рівня довіри до платформи. Системи відгуків і рейтингів дозволяють користувачам оцінювати продавців і ділитися досвідом, що допоможе іншим користувачам приймати обґрунтовані рішення.

Шостим критерієм виступає технічна стабільність платформи. Вона передбачає відсутність технічних збоїв та помилок, а також швидкість реагування на можливі проблеми. Технічно стабільний сервіс гарантує безперебійну роботу та позитивний користувацький досвід. Швидке вирішення технічних питань та оновлення системи підвищують задоволеність користувачів і зміцнюють їхню лояльність до платформи.

Таким чином, оцінка ефективності існуючих рішень базується на комплексному аналізі користувацького досвіду, функціональності, залученості користувачів, конверсії, безпеки та технічної стабільності. Ретельний аналіз цих критеріїв дозволяє виявити сильні та слабкі сторони онлайн-платформ для підбору автомобілів і визначити напрямки для їх подальшого вдосконалення.

2 РОЗРОБКА МЕТОДИКИ ПІДБОРУ АВТОМОБІЛІВ З ВИКОРИСТАННЯМ РЕКОМЕНДАЦІЙНИХ АЛГОРИТМІВ

2.1 Постановка задачі

Зростаючий асортимент автомобілів різних марок, моделей, комплектацій та цінових категорій значно ускладнює процес вибору для потенційних покупців. Вибір оптимального автомобіля вимагає аналізу численних факторів, таких як технічні характеристики, економічна ефективність, безпека, дизайн, екологічність та інші індивідуальні критерії. Цей процес часто займає значний час і потребує значних зусиль, що може призводити до фрустрації споживачів та навіть відмови від покупки.

Однією з базових проблем, з якою зіштовхуються покупці при виборі автомобіля, є інформаційне перенавантаження. Велика кількість доступних варіантів та необхідність врахування численних параметрів роблять процес вибору складним та трудомістким. Традиційні методи, такі як консультації з продавцями або самостійний пошук інформації в інтернеті, часто не забезпечують необхідної ефективності та точності, що може призводити до неправильного вибору або затримок у прийнятті рішення.

Крім того, ринок автомобілів постійно розвивається, з'являються нові моделі та технології, що ще більше ускладнює завдання покупцям. Висока конкуренція між виробниками змушує їх пропонувати все більш інноваційні та різноманітні продукти, що потребує від споживачів постійного оновлення знань та адаптації до нових умов. У таких умовах виникає необхідність у створенні ефективних інструментів, які могли б автоматизувати процес вибору автомобіля, забезпечуючи швидкість, точність та персоналізований підхід до кожного користувача.

Вдосконалення технологій машинного навчання та штучного інтелекту забезпечує нові шляхи розв'язання цієї проблеми. Застосування рекомендаційних

алгоритмів дає змогу виконати аналіз значних обсягів даних про автомобілі, їхні характеристики, відгуки користувачів та ринкові тенденції, що забезпечує генерацію персоналізованих рекомендацій для кожного споживача. Це не лише скорочує час, необхідний для вибору автомобіля, але й підвищує ймовірність задоволеності покупця кінцевим результатом.

Для автентифікації процесу вибору автомобіля було вирішено розробити методику підбору, яка використовуватиме рекомендаційні алгоритми. Така методика повинна забезпечити швидке та точне надання рекомендацій, з урахуванням індивідуальних потреб та вподобань кожного користувача. Це дозволить споживачам швидко знаходити автомобілі, що найкраще відповідають їхнім вимогам, зменшуючи час та зусилля, необхідні для прийняття рішення.

Ключовим етапом розробки методики є збір та обробка масштабних обсягів даних, що включають інформацію про різні моделі автомобілів, їхні технічні характеристики, відгуки користувачів, ринкові тенденції та інші релевантні параметри. Для цього необхідно створити комплексну базу даних, яка забезпечить доступ до актуальної та детальної інформації. Застосування сучасних інструментів для аналізу та візуалізації даних дозволить ефективно обробляти ці дані та забезпечити їхню високу якість[11].

Розробка ефективних алгоритмів машинного навчання є ще одним важливим аспектом створення методики. Алгоритми повинні бути здатні аналізувати зібрані дані, виявляти закономірності та взаємозв'язки між різними параметрами автомобілів, а також враховувати індивідуальні уподобання користувачів. Це дозволить генерувати точні та релевантні рекомендації, що відповідають потребам кожного споживача.

Інтеграція розроблених алгоритмів у зручну для користувача систему є критично важливою для забезпечення високої ефективності методики. Система повинна бути інтуїтивно зрозумілою та легкою у використанні, забезпечуючи швидкий доступ до рекомендацій та можливість оперативного коригування параметрів вибору. Оптимізація алгоритмів для роботи в реальному часі та мінімізація часу відгуку системи також є важливими аспектами, які впливають на

загальну ефективність методики[12].

Важливо також врахувати можливість масштабування системи для подальшого розширення функціональності та адаптації до змінних потреб ринку. Розроблена методика повинна бути гнучкою та адаптивною, дозволяючи інтегрувати нові моделі автомобілів, оновлювати базу даних та вдосконалювати алгоритми на основі отриманих результатів та зворотного зв'язку від користувачів.

Одним із ключових аспектів розробки методики є її тестування та постійне вдосконалення. Перед впровадженням системи необхідно провести ретельне тестування для виявлення та виправлення можливих помилок та недоліків. Тестування дозволить підвищити надійність та ефективність роботи системи, забезпечуючи її готовність до реального використання. Залучення експертів у галузі автомобільної промисловості та аналізу даних допоможе забезпечити високу якість рекомендацій та адаптивність системи до різних сценаріїв використання.

Результатом успішної розробки методики стане інноваційний інструмент, який поєднує в собі високу ефективність, точність рекомендацій та зручність використання. Така система не лише зменшить час, необхідний для вибору автомобіля, але й підвищить задоволеність користувачів процесом покупки, сприяючи підвищенню лояльності клієнтів та успішності бізнесу на конкурентному ринку автомобілів. Крім того, впровадження рекомендаційних алгоритмів дозволить компаніям краще розуміти потреби своїх клієнтів, що відкриє нові можливості для персоналізації пропозицій та покращення сервісу.

Таким чином, постановка задачі полягає у створенні методики підбору автомобілів, яка використовує передові рекомендаційні алгоритми для оптимізації процесу вибору. Це дозволить значно скоротити час, необхідний для вибору автомобіля, та покращити загальний користувацький досвід, відповідаючи сучасним вимогам ринку та потребам споживачів.

2.2 Основні формули та критерії традиційного методу підбору автомобіля та методу підбору автомобіля на основі штучного інтелекту

Традиційний метод підбору автомобіля.

Традиційний метод базується на використанні зважених критеріїв для оцінки автомобілів[13]. Кожному критерію K_i присвоюється ваговий коефіцієнт W_i , що відображає його важливість.

Кроки методу:

Визначення критеріїв та вагових коефіцієнтів:

Критерії:

- Ціна (C)
- Пробіг (P)
- Вік автомобіля (V)
- Витрати пального (F)
- Надійність (R)
- Комфорт (K)

Вагові коефіцієнти (W_i):

- $W_C, W_P, W_V, W_F, W_R, W_K$

Нормалізація критеріїв:

Для забезпечення порівнянності критеріїв їх значення нормалізуються до діапазону від 0 до

- Для критеріїв, де менше значення краще (2.1):

$$K_{i_{\text{норм}}} = \frac{K_{\text{макс}} - K_i}{K_{\text{макс}} - K_{\text{мін}}}. \quad (2.1)$$

- Для критеріїв, де більше значення краще (2.2):

$$K_{i_{\text{норм}}} = \frac{K_i - K_{\text{мін}}}{K_{\text{макс}} - K_{\text{мін}}}. \quad (2.2)$$

Розрахунок загального рейтингу автомобіля (2.3):

$$S = \sum_i K_{i_{\text{норм}}} \times W_i . \quad (2.3)$$

Показник часу для традиційного методу (2.4):

$$T_{\text{традиц}} = n * t_0 , \quad (2.4)$$

де n — кількість критеріїв;

t_0 — час на обробку одного критерія.

Метод підбору автомобіля, що базується на штучному інтелекті.

Саме цей метод застосовує алгоритми машинного навчання, щоб врахувати складні взаємозв'язки між критеріями та підвищувати точність оцінювання.

Основні компоненти методу.

Побудова моделі машинного навчання.

Модель навчається на історичних даних та визначає вагові коефіцієнти β_i для кожного критерію.

Розрахунок прогнозованого рейтингу автомобіля.

Рейтинг Z обчислюється за формулою (2.5):

$$Z = \sum_i \beta_i \times K_{i_{\text{авт}}} , \quad (2.5)$$

де β_i — вагові коефіцієнти для кожного критерію;

$K_{i_{\text{авт}}}$ — нормалізовані значення критеріїв (ціна, пробіг, вік автомобіля тощо).

Врахування нелінійних взаємозв'язків:

Модель може включати нелінійні функції для взаємодії між критеріями, що покращує точність прогнозу.

Показник часу для методу ШІ (2.6):

$$T_{\text{ШІ}} = T_{\text{навч}} + T_{\text{прогн}} , \quad (2.6)$$

де $T_{\text{навч}}$ – час необхідний для навчання моделі;

$T_{\text{прогн}}$ – час прогнозування для одного автомобіля.

Експериментом із комплексними числами.

Час для звичайного методу підбору:

Час на обробку одного критерія: $t_0 = 1$ хвилина.

Кількість критеріїв: $n=6$.

Час обробки для одного автомобіля:

$$T_{\text{традиц}} = t_0 \times n = 1 \times 6 = 6 \text{ хвилин.}$$

Загальний час для двох автомобілів:

$$T_{\text{заг}} = 6 \times 2 = 12 \text{ хвилин.}$$

Час для методу з використанням штучного інтелекту (2.7):

$$T_{\text{навч}} = a \cdot n^2 + b, \quad (2.7)$$

де змінні a і b є параметрами, які використовуються для обчислення часу навчання моделі III (time_train).

У традиційному методі час на обробку шести критеріїв для одного автомобіля становить 6 хвилин, а загальний час для двох автомобілів — 12 хвилин. Для методу штучного інтелекту час навчання моделі складає 60 хвилин, а прогнозування для одного автомобіля — 1 хвилина. Це дозволяє оцінити значну різницю у часі виконання завдань між двома підходами, як це показано в Таблиці 2.1.

Таблиця 2.1.

Параметр	Традиційний метод	Метод III (перший запуск)	Метод III (наступні запуски)
Загальний час (2 авто)	12 хвилин	61 хвилина	1 хвилина
Час на авто	6 хвилин	0.5 хвилини	0.5 хвилини

Таблиця порівняння показників часу для різних методів.

Традиційний метод швидший при невеликій кількості автомобілів, якщо модель III ще не навчена. Метод III ефективніший за часом при повторному використанні навченої моделі, особливо при великій кількості автомобілів.

2.3 Збір та обробка великих обсягів даних

Збір та обробка великих обсягів даних

Велика кількість інформації включає в себе збір і аналіз обсягів даних, які занадто великі або складні для традиційних баз даних і програмного забезпечення. Аналіз цих даних вимагає використання потужних систем з обробки паралельних обчислень.

Фази обробки:

- -збір різноджерельної інформації, включаючи Інтернет речей, соціальні мережі, комерційні операції тощо;
- -видалення помилок, дублікатів і нерелевантної інформації;
- -об'єднання даних в єдину, доступну базу;
- -використання статистичних методів і машинного навчання для отримання інсайтів з даних.

Зважаючи на важливість ефективного збору та обробки даних у нинішньому світі, додатково можна розглянути кілька аспектів, що впливають на процес збору та обробки певних обсягів інформації. Перш за все, необхідно враховувати різноманітність інформаційних джерел та їх форматів. Сучасна інформаційна ландшафт постійно змінюється, і разом з ним зростає кількість джерел, з яких можна отримувати дані. Окрім традиційних джерел, таких як бази даних і веб-сервери, важливо враховувати нові джерела, такі як датчики Інтернету речей, соціальні мережі, медичні пристрої та інші.

Крім того, створення методології збору даних, що забезпечує їх повноту і актуальність, є ключовим етапом. Наприклад, у випадку з даними з соціальних мереж, важливо враховувати широкий спектр можливих джерел інформації, а

також швидкі зміни в трендах та популярності, щоб забезпечити актуальність даних.

Додатково, важливо розглядати аспекти безпеки даних під час їх збору та обробки. Збір та передача великих обсягів інформації може створювати потенційні загрози щодо конфіденційності та цілісності даних. Тому важливо вживати заходів для захисту даних на кожному етапі їх життєвого циклу, починаючи від збору і закінчуючи аналізом та зберіганням.

Загалом, успішна обробка великих інформаційних обсягів вимагає комплексного підходу, який враховує різноманітність джерел і форматів даних, розроблення методології збору, а також передбачення безпеки та конфіденційності даних. Цей підхід дозволяє максимально використовувати потенціал великих обсягів даних для прийняття обґрунтованих рішень та виявлення цінних інсайтів.

2.4 Інструменти та технології для роботи з великими даними

Робота з великими даними вимагає використання спеціалізованих інструментів і технологій, які забезпечують зберігання, обробку та аналіз великих обсягів структурованих і неструктурованих даних. Основні технології в цій сфері мають свої унікальні характеристики, які дозволяють вибирати найбільш відповідний інструмент залежно від конкретних завдань.

Hadoop є комплексною екосистемою для роботи з великими даними, яка включає такі ключові компоненти, як HDFS (Hadoop Distributed File System) для розподіленого зберігання даних і MapReduce для їхньої обробки. Ця технологія відома своєю стійкістю до збоїв та можливістю горизонтального масштабування, що робить її оптимальною для обробки великих пакетних завдань[14].

Apache Spark є потужною платформою, яка забезпечує надзвичайно швидку обробку даних завдяки можливості виконання операцій у пам'яті. Spark включає в себе широкий діапазон завдань, включаючи потокову передачу даних, машинне навчання, графові обчислення та роботу з базами даних. Завдяки своїй

універсальності та високій продуктивності[15] Spark активно використовується для завдань, які вимагають швидких результатів, наприклад, в аналітиці реального часу чи в рекомендаційних системах.

NoSQL бази даних, такі як MongoDB та Cassandra, відрізняються своєю здатністю працювати з неструктурованими та напівструктурованими даними[16]. Ці системи особливо ефективні для додатків, що вимагають гнучкості в схемах даних, наприклад, у веб-розробці або при зберіганні логів користувацької активності. MongoDB пропонує просту інтеграцію з іншими технологіями та можливість виконання складних запитів, тоді як Cassandra забезпечує високу доступність і горизонтальне масштабування.

Для детального порівняння цих технологій з точки зору продуктивності, масштабованості та застосування, зверніться до Таблиці 2.2, яка містить ключові аспекти, що допоможуть зрозуміти їх переваги та оптимальні сценарії використання.

Таблиця 2.2

Таблиця порівняння основних технологій.

Критерій	Hadoop MapReduce	Apache Spark	NoSQL бази даних
Швидкість	Повільніше, обчислення виконуються на дисках	Швидше, можливість виконання в пам'яті	Залежить від типу бази даних; швидкий доступ до неструктурованих даних
Застосування	Великі пакетні завдання, стійке зберігання	Машинне навчання, потокові обчислення	Широкий спектр застосувань, включно з веб-додатками, кешуванням та великими даними
Масштабування	Лінійне, кластери	Лінійне, можливість виконання у кластерах та хмарах	Горизонтальне масштабування, легко адаптується до великих обсягів даних

Продовження таблиці 2.2

Таблиця порівняння основних технологій.

Критерій	Hadoop MapReduce	Apache Spark	NoSQL бази даних
Використання	Простіше API для пакетних процесів	Складніше API, але більш гнучке для різних видів завдань	Гнучкі схеми, проста інтеграція з різними додатками

2.5 Аналіз та вибір алгоритмів з подальшою їх адаптацією

Огляд існуючих алгоритмів машинного навчання

Аналіз алгоритмів машинного навчання, що можуть бути використані для обробки даних у автомобільній індустрії, повинен охоплювати широкий спектр технік, від класичних методів до сучасних підходів глибокого навчання[17]. Наприклад:

-лінійні моделі (логістична регресія): ці моделі ефективні для задач, де потрібно оцінити ймовірність настання події (наприклад, чи буде покупець задоволений певною моделлю авто). Логістична регресія проста у використанні і інтерпретації, але може не впоратися зі складними нелінійними залежностями;

-дерева рішень і ансамблі дерев (випадковий ліс, градієнтний бустинг): ці алгоритми добре працюють з наборами даних, що містять числові та категорійні дані, і здатні виявляти складні взаємозв'язки між атрибутами. Вони особливо корисні для класифікації та регресії в контексті великої кількості вхідних параметрів. При побудові дерева рішень часто використовується індекс Джині для оцінки якості розбиття.

-нейронні мережі (особливо конволюційні та рекурентні мережі): ці моделі ідеально підходять для обробки зображень (наприклад, аналіз зовнішнього вигляду автомобіля) та послідовних даних (наприклад, часові ряди про динаміку

цін). Вони потребують великої кількості обчислювальних ресурсів та значних обсягів даних для навчання;

-методи зниження розмірності (PCA, t-SNE): ці алгоритми корисні для візуалізації високовимірних даних та можуть допомогти виявити приховані залежності та групи у даних.

Рис. 2.1 Візуалізація демонструє модель машинного навчання, застосовану до наявних даних. Ліва частина ілюстрації представляє вихідні спостереження, які містять три змінні: висоту, ширину та форму. Фігури, що використовуються для відображення даних, включають зірочки, хрестики та трикутники.

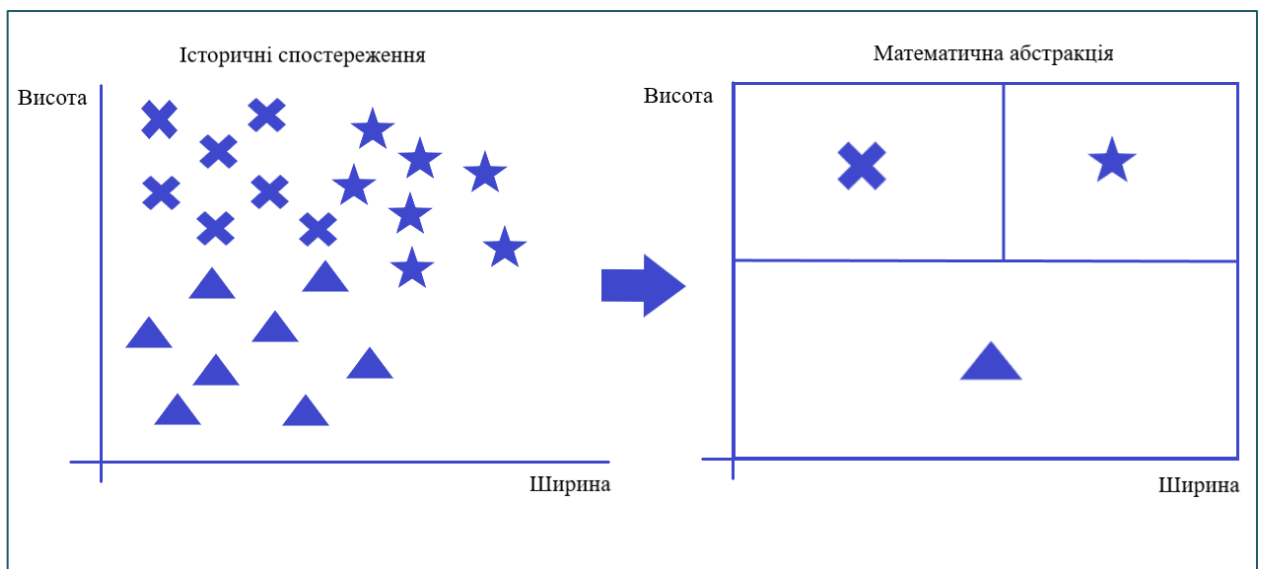


Рис. 2.1 Основний процес машинного навчання

Застосування ключових параметрів для оцінки ефективності алгоритмів

Для адекватної оцінки методів машинного навчання важливо визначити ключові метрики. Серед них:

1. Точність
2. Швидкість обробки
3. Здатність до масштабування
4. Робастність

Розробка комплексного підходу

Розробка комплексного підходу, який інтегрує регресійний аналіз і глибокі нейронні мережі, дозволяє виявляти складні зв'язки в даних, які можуть залишатися непоміченими при застосуванні традиційних методів.

Інтеграція регресійного аналізу та глибоких нейронних мереж:

Регресійний аналіз [18] використовується для ідентифікації та кількісного оцінювання зв'язків між змінними, наприклад, ціною автомобіля і його характеристиками.

Глибокі нейронні мережі використовуються для моделювання складних патернів і зв'язків, які можуть існувати в даних на рівні, недоступному для простих регресійних моделей.

Підхід:

1.1 Спочатку використовується регресійний аналіз для визначення основних тенденцій і відносин.

1.2 Далі застосовуються нейронні мережі для детальнішого аналізу та виявлення складніших патернів у даних.

1.3 Ці моделі можуть бути з'єднані в каскадній або паралельній структурі для забезпечення глибокого та всебічного аналізу даних.

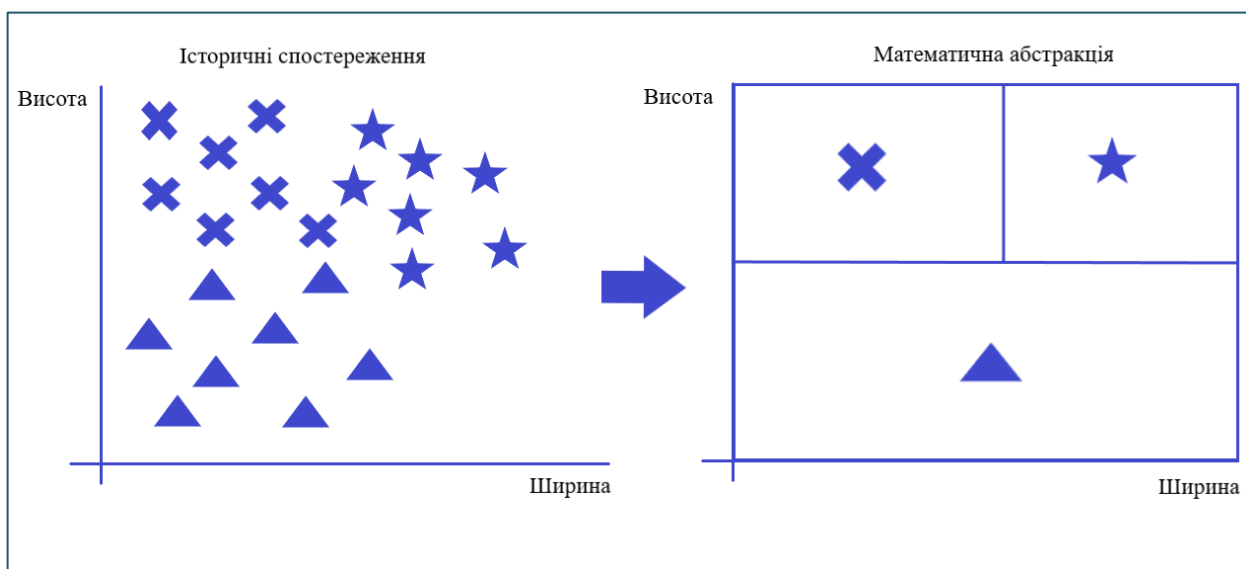


Рис. 2.2 Порівняння результатів роботи регресійної і класифікаційної моделей

Ліва частина зображення відображає класифікацію, де цільова змінна — це форма спостереження, представлена як категоріальна змінна. Права частина ілюструє регресію, де цільова змінна є числовою. Хоча правила прийняття рішень для обох випадків можуть бути ідентичними, їхня інтерпретація суттєво відрізняється.

Ця інтеграція забезпечує більш глибокий аналіз та вищу точність прогнозування, що є значущим для ефективності рекомендаційної системи у проекті.

2.6 Експериментальні дослідження та оцінка ефективності методики

Для проведення експериментальних досліджень у рамках роботи було обрано програмне забезпечення AnyLogic, оскільки воно надає можливість моделювати складні системи, що включають різні типи процесів, а також дозволяє візуалізувати роботу рекомендаційних алгоритмів у реальному часі. Завдяки універсальності цього ПЗ, було реалізовано моделювання процесу підбору автомобілів. Інструментарій AnyLogic забезпечив гнучкість у налаштуванні експериментів і аналізі отриманих результатів, що дозволило провести серію симуляцій для оцінки ефективності запропонованої методики[19].

Опис параметрів та блоків модуляції:

1. Основні змінні моделі

Цей блок містить змінні, які визначають стан системи та використовуються для відображення динаміки моделювання:

- Num_cars — кількість автомобілів у системі. Початкове значення 10, змінюється під час моделювання залежно від обробки автомобілів.
- process_car_traditional і process_car_AI_FIRST/NEXT — змінні для відстеження кількості автомобілів, оброблених традиційним методом та за допомогою штучного інтелекту на різних етапах.
- time_criteria, time_predict, time_train — змінні, пов'язані з часом аналізу критеріїв, прогнозування та навчання ШІ.

- `var_AI_First/NEXT`, `var_traditional` — допоміжні змінні для відображення станів або характеристик процесу роботи ІІІ та традиційних підходів.

2. Динамічна подія

Подія виконує ключову функцію в моделюванні, змінюючи значення змінних у часі.

Вона автоматично оновлює:

- `process_car_traditional` — збільшує кількість автомобілів, оброблених традиційним методом.
- `process_car_AI_FIRST/NEXT` — збільшує кількість автомобілів, оброблених ІІІ на першому або наступних етапах. Подія відображає прив'язку до часу, дозволяючи моделі працювати в режимі динамічного оновлення.

3. Параметри для обчислень

Змінні `a` і `b` є параметрами, що використовуються для обчислення часу навчання моделі ІІІ (`time_train`).

`a` — коефіцієнт, який враховує складність взаємозв'язків між критеріями (час на обчислення залежностей для одного критерію);

`b` — базовий час підготовки моделі, який не залежить від кількості критеріїв (наприклад, ініціалізація моделі, завантаження даних тощо).

Цей час є важливим для оцінки ефективності моделі ІІІ в системі.

4. Графічне відображення

Графік ілюструє динаміку змін змінних у часі, надаючи візуальне уявлення про роботу моделі:

- Синя лінія: кількість автомобілів, оброблених традиційним методом.
- Помаранчева лінія: оброблені автомобілі традиційним методом.
- Помаранчева і рожева лінії: оброблені автомобілі ІІІ на першому та наступних етапах відповідно.

Графік дозволяє аналізувати ефективність роботи системи (традиційний підхід vs ІІІ) Рис 2.3.

Ця структура забезпечує зручний поділ на основні компоненти моделі для

чіткого розуміння функціоналу та взаємозв'язків у системі.

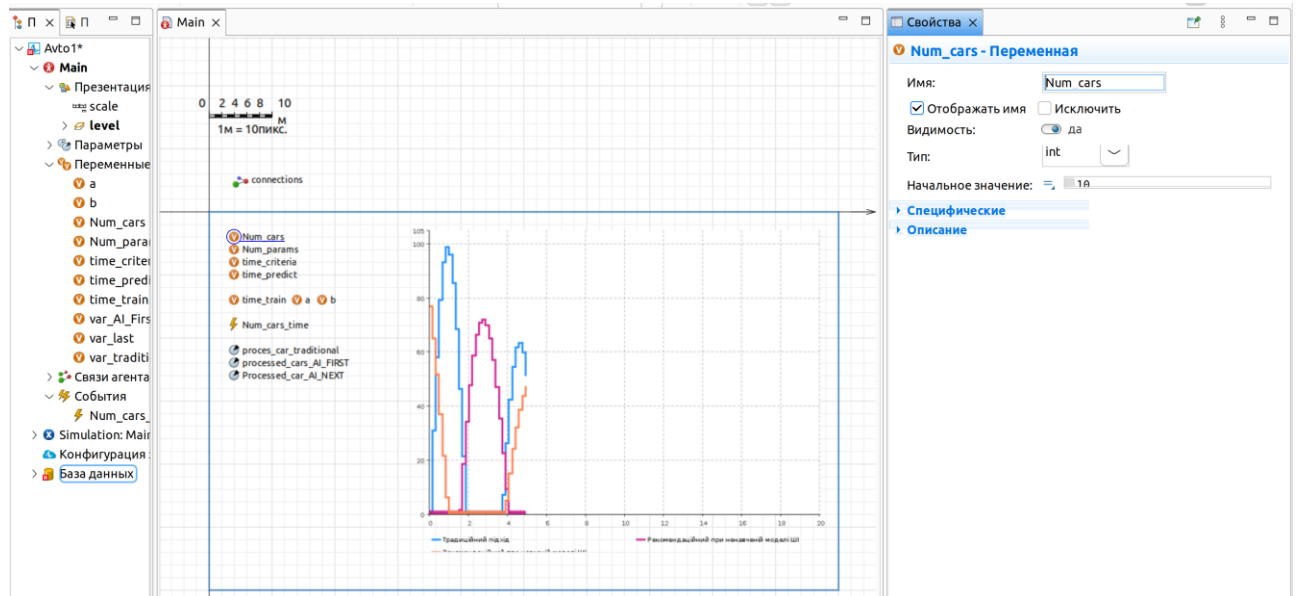


Рис. 2.3 Вікно модуляції Any Logic

Проведено декілька модуляцій з різними початковими параметрами, що дало можливість встановити важливість параметрів та те який з них є більш значний для поставленого завдання та як це впливає на процес підбору.

В результаті проведених експериментів отримано наступні моделі:

Симуляція 1: результати моделювання для невеликої кількості автомобілів та параметрів рис 2.4.

Змінювані параметри:

Num_params = 6

Num_cars = 10

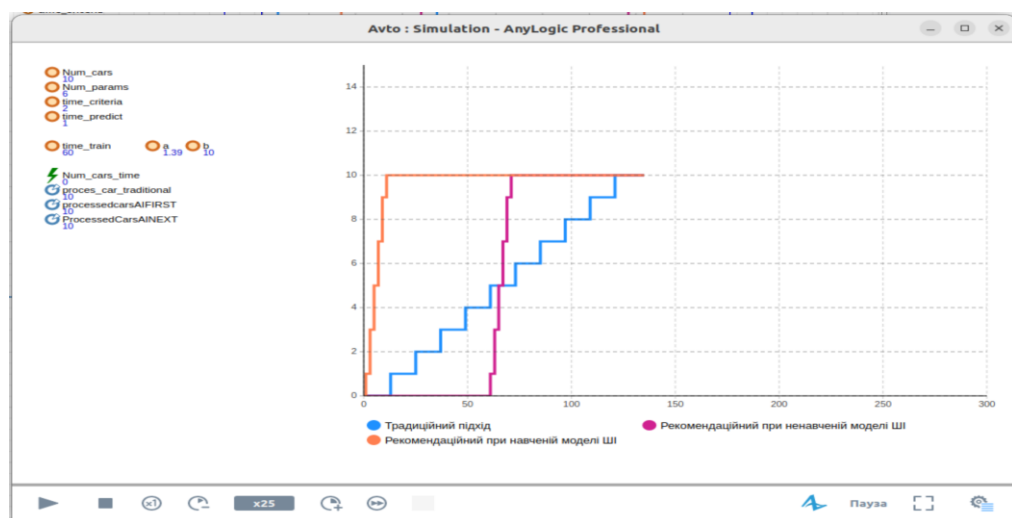


Рис. 2.4 Базовий набір параметрів

Симуляція 2: результати моделювання для невеликої кількості автомобілів зі збільшеною кількістю параметрів рис 2.5.

Змінювані параметри:

Num_params = 15

Num_cars = 10

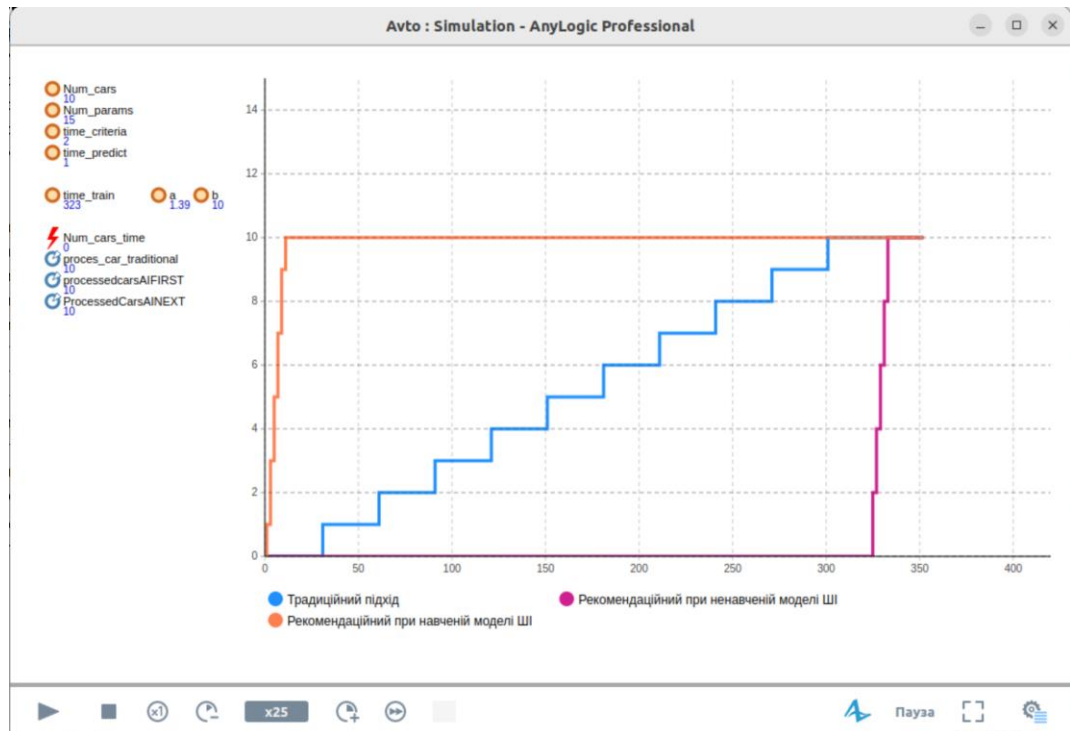


Рис. 2.5 Складний підбір

Симуляція 3: результати моделювання для збільшеної кількості автомобілів при стандартній кількості параметрів рис 2.6.

Змінювані параметри:

Num_params = 6

Num_cars=50

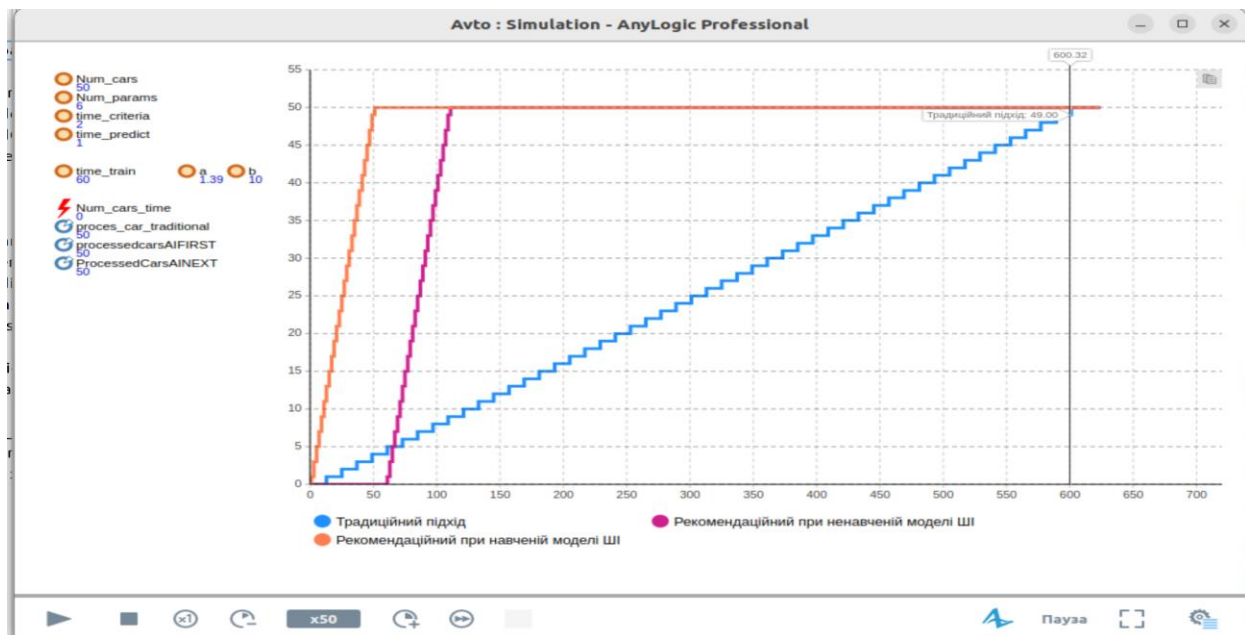


Рис. 2.6 Масовий підбір авто

Симуляція 4: результати моделювання для збільшеної кількості автомобілів і параметрів у найскладнішому сценарії рис 2.7.

Змінювані параметри:

Num_params = 15

Num_cars = 50

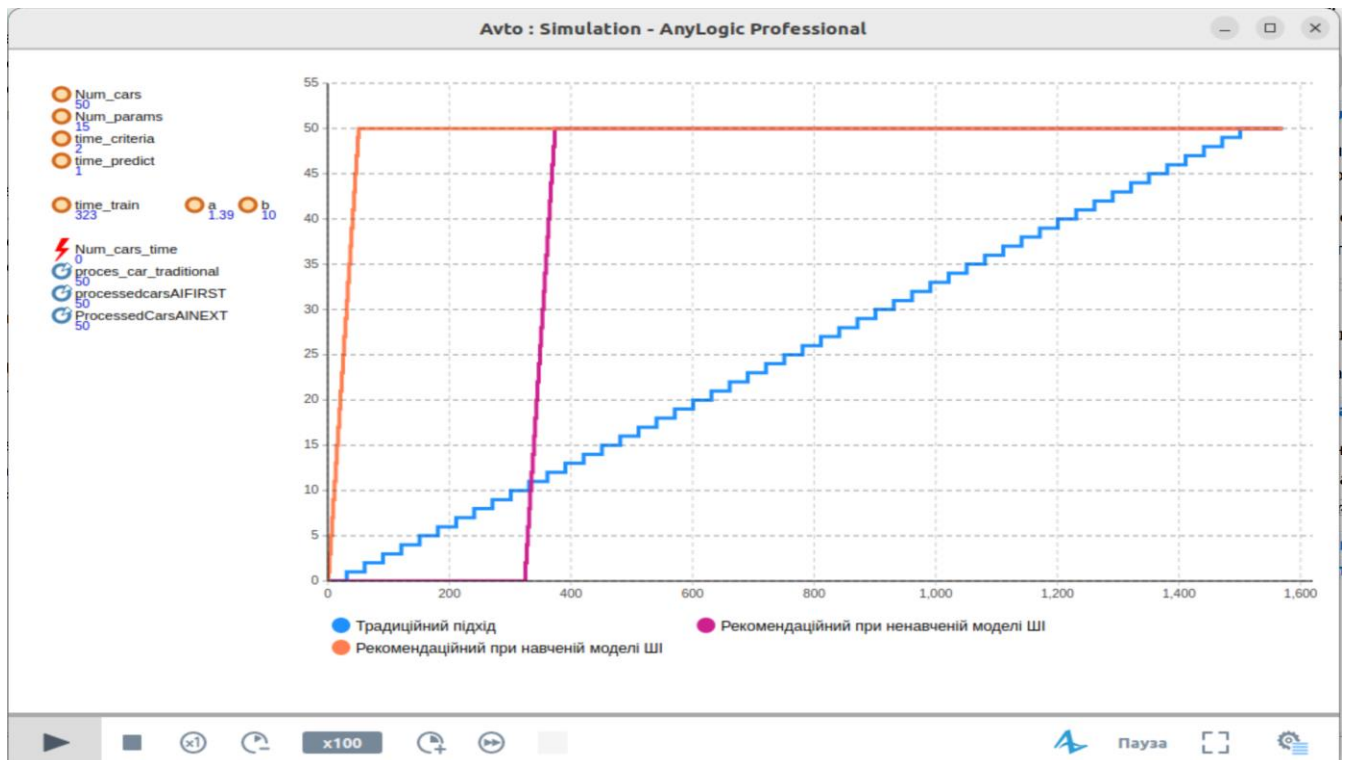


Рис. 2.7 Комплексний масовий підбір

2.7 Оцінка впливу на користувацький досвід

Оцінка запропонованої методики підбору автомобілів за допомогою навченої моделі штучного інтелекту (ШІ) на кінцевий досвід користувача враховує швидкість, точність та якість рекомендацій. Швидкість отримання результатів є критичним фактором для задоволення користувача[20]. Навчена модель ШІ забезпечує майже миттєву генерацію рекомендацій, особливо у випадках з невеликою кількістю параметрів та автомобілів, як показано в симуляції 1. Навіть при збільшенні кількості параметрів та обсягу бази даних, як у симуляції 4, час обробки залишається значно меншим, ніж у традиційному підході. Це дозволяє користувачам швидко отримувати релевантну інформацію без затримок.

Якість рекомендацій безпосередньо впливає на довіру та задоволеність користувача. Завдяки можливості враховувати складні критерії та пріоритети користувача, навчена модель ШІ надає більш точні та персоналізовані рекомендації. Модель здатна ефективно фільтрувати та ранжувати автомобілі за значущістю для користувача, що підвищує ймовірність вибору оптимального варіанту[21].

Поліпшення взаємодії з системою сприяє позитивному користувацькому досвіду. Інтуїтивний інтерфейс, створений завдяки швидкій та точній роботі моделі, робить систему більш зручною та простою у використанні. Модель ефективно обробляє запити незалежно від їх складності, що робить систему гнучкою та адаптивною до вимог різних користувачів.

Традиційний підхід має суттєві обмеження, які негативно впливають на користувацький досвід. Значний час виконання завдань призводить до затримок, що може викликати фрустрацію у користувачів. Обмежені можливості врахування індивідуальних потреб знижують релевантність рекомендацій, а відсутність персоналізації та гнучкості робить систему менш привабливою.

Ненавчена модель ШІ пропонує певні переваги над традиційним підходом, але має свої недоліки. Вона працює швидше за традиційний метод, але повільніше за навчений ШІ. Без попереднього навчання модель не може повною мірою враховувати складні запити, що знижує якість рекомендацій. Менша адаптивність ненавченої моделі робить її менш ефективною у персоналізації рекомендацій.

Запропонована методика з використанням навченої моделі ШІ позитивно впливає на користувацький досвід. Значне скорочення часу очікування дозволяє користувачам отримувати результати швидше, підвищуючи їх задоволеність та знижуючи ймовірність відмови від використання системи. Точні та релевантні рекомендації допомагають приймати більш обґрунтовані рішення, а інтуїтивний інтерфейс та адаптивність системи роблять процес підбору автомобілів приємним та ефективним. Впровадження навченої моделі ШІ значно поліпшує загальний досвід користувача, роблячи систему більш конкурентоспроможною та привабливою на ринку. Це підтверджує доцільність використання сучасних технологій ШІ для вирішення задач підбору автомобілів та інших складних інформаційних систем.

2.8 Оцінка ефективності методики та порівняння з існуючими підходами

Метою даної оцінки є визначення ефективності запропонованої методики підбору автомобілів за допомогою навченої моделі штучного інтелекту (ШІ) у порівнянні з традиційним підходом та ненавченою моделлю ШІ. Оцінка базується на числових показниках, отриманих з чотирьох симуляцій, які відображають різні сценарії використання системи.

Для проведення оцінки було здійснено чотири симуляції з варіацією двох ключових параметрів: кількості параметрів, які є характеристиками автомобілів, що враховуються при підборі, та кількості автомобілів, що визначає обсяг бази даних для підбору.

У всіх симуляціях було встановлено, що навчена модель ІІІ демонструє значно вищу ефективність у порівнянні з іншими підходами[22]. Вона стабільно забезпечує швидке виконання завдань та високу точність рекомендацій, незалежно від кількості параметрів чи автомобілів. У симуляціях з невеликою кількістю автомобілів навчена модель ІІІ перевершує традиційний підхід та ненавчену модель навіть при мінімальних обсягах даних, демонструючи швидше виконання завдань та кращу точність. При збільшенні кількості параметрів час виконання завдань зростає у всіх підходах, але навчена модель ІІІ все ще забезпечує найкращу продуктивність. У симуляціях з великою кількістю автомобілів, при зростанні обсягу бази даних, навчена модель ІІІ зберігає високу ефективність, тоді як традиційний підхід демонструє значне збільшення часу обробки. У найскладніших сценаріях з максимальними значеннями параметрів та автомобілів навчена модель ІІІ показує стабільність та масштабованість, залишаючись найефективнішим рішенням.

Аналіз результатів свідчить, що навчена модель ІІІ послідовно демонструє найвищу продуктивність, особливо при збільшенні складності завдань. Традиційний підхід виявляє експоненціальне зростання часу виконання зі збільшенням кількості параметрів та автомобілів, що робить його неефективним для сучасних вимог. Ненавчена модель ІІІ пропонує проміжну продуктивність, але значно відстає від навченої моделі. У плані точності рекомендацій навчена модель ІІІ забезпечує високу точність завдяки оптимізованим алгоритмам та здатності враховувати складні критерії. Ненавчена модель ІІІ має нижчу точність через відсутність адаптації до специфічних вимог, а традиційний підхід обмежений у можливостях обробки складних критеріїв, що негативно впливає на якість рекомендацій.

Щодо масштабованості та універсальності, навчена модель ІІІ демонструє високу масштабованість, підтримуючи ефективність при збільшенні обсягів даних. Традиційний підхід не має масштабованості і стає непрактичним для великих систем. Ненавчена модель ІІІ має обмежену масштабованість та універсальність у порівнянні з навченою моделлю[23].

Аналіз симуляцій показує, що навчена модель ІІІ перевершує інші методи за всіма ключовими показниками: швидкість, точність та масштабованість. Традиційний підхід виявляється найменш ефективним, особливо при збільшенні складності завдань. Ненавчена модель ІІІ, хоча і перевершує традиційний підхід у деяких аспектах, все ж значно поступається навченій моделі ІІІ.

Результати чотирьох симуляцій однозначно свідчать про переваги навченої моделі ІІІ над традиційним підходом та ненавченою моделлю. Навчена модель не тільки забезпечує значне скорочення часу обробки запитів, але й підвищує якість та релевантність рекомендацій. Це підтверджує доцільність впровадження такої методики в реальних умовах, особливо для систем з великим обсягом даних та складними вимогами користувачів. Таким чином, навчена модель ІІІ є ключовим елементом для підвищення ефективності та конкурентоспроможності сучасних систем підбору автомобілів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕКОМЕНДАЦІЙ

3.1 Використані програмні засоби

3.1.1 Мова програмування та середовище розробки

Для розробки інноваційної системи було обрано мову програмування Python завдяки її популярності, широким можливостям та багатому вибору бібліотек для машинного навчання, штучного інтелекту та веб-розробки. Python забезпечує високу швидкість розробки та має активну спільноту, що сприяє швидкому вирішенню технічних проблем і впровадженню новітніх технологій. Поєднання простоти, елегантності та потужності Python робить її ідеальним інструментом для створення ефективної та унікальної системи.

Як веб-фреймворк було обрано Django — високорівневий інструмент для розробки веб-додатків на Python. Django надає засоби для швидкого створення безпечних, масштабованих та інтуїтивно зрозумілих веб-додатків. Він містить багато вбудованих функцій, таких як адмін-панель, авторизація та аутентифікація користувачів, що робить процес розробки набагато легшим. Архітектура Model-View-Template (MVT) в Django дозволяє структурувати та розробляти проєкт у модульний спосіб, що відповідає вимогам замовника [24].

У процесі розробки проєкту було використано інтегроване середовище розробки PyCharm, що є ефективним інструментом для написання, тестування та налагодження коду. PyCharm забезпечує інтерфейс, що спрощує всі аспекти розробки; його функціональні можливості автодоповнення, рефакторингу та інтеграції з системами контролю версій підвищують продуктивність і зручність. За допомогою цього середовища ми ефективно керуємо проєктом та можемо зосередитися на досягненні оптимальних результатів.

Таким чином, вибір мови програмування Python, веб-фреймворку Django та середовища розробки PyCharm є оптимальною комбінацією для успішної реалізації системи. Використання цих інструментів дозволяє створити продукт

високої якості, який задовольняє вимоги користувачів. Команда проєкту спрямована на забезпечення якісної підтримки та вирішення завдань, що стоять перед нею. Розроблена система є надійною, масштабованою та інтуїтивно зрозумілою для користувачів, що відзначається високою якістю та функціональністю.

3.1.2 Оптимальні інструменти для роботи з даними

Для обробки, зберігання та аналізу даних у проєкті було використано широкий спектр інструментів, які забезпечили надійність, продуктивність і високу точність роботи системи. Нижче наведено розширений опис кожного з них.

PostgreSQL — це реляційна система керування базами даних з відкритим кодом, яка відіграла ключову роль у зберіганні структурованих даних. Вона базується на реляційній моделі, що дозволяє ефективно зберігати інформацію про автомобілі, їх технічні характеристики, рейтинги та взаємозв'язки між ними. Надійність і масштабованість PostgreSQL забезпечується підтримкою транзакцій ACID, що гарантує консистентність даних навіть при великих обсягах обробки. Крім того, інтеграція з Django ORM спростила процес виконання CRUD-операцій та складних запитів, оскільки дозволила взаємодіяти з базою даних через об'єктно-орієнтовану модель[25].

NumPy — бібліотека для наукових обчислень, яка використовувалася для обробки числових даних. Вона надає можливість працювати з багатовимірними масивами та матрицями, що є необхідним для представлення ознак автомобілів. Ефективне виконання математичних операцій, включаючи нормалізацію, трансформацію та обчислення агрегатів, забезпечило підготовку даних перед передачею їх у модель машинного навчання. NumPy відіграла важливу роль у стандартизації та обробці числових ознак, що підвищило точність моделі.

Pandas — це потужна бібліотека для роботи з табличними даними, яка дозволяє ефективно обробляти великі обсяги інформації. Завдяки структурі

DataFrame, вона значно спрощує маніпулювання та трансформацію даних, що є важливим етапом підготовки даних для моделювання. Pandas забезпечує зручний імпорт даних з різних форматів, таких як CSV, Excel, JSON або SQL, що надає гнучкість при роботі з різноманітними джерелами. Бібліотека також активно використовувалася для очищення даних, видалення пропусків, фільтрації та агрегації, що допомогло покращити якість даних перед їх подальшим аналізом.[26]

Scikit-learn — бібліотека для машинного навчання, яка забезпечила весь спектр необхідних інструментів для побудови та оцінювання моделей. Вона використовувалася для попередньої обробки даних, зокрема для нормалізації та стандартизації ознак із використанням StandardScaler. Scikit-learn підтримує широкий спектр алгоритмів машинного навчання, таких як лінійна регресія, кластеризація та методи ансамблів, що дозволило вибрати оптимальну модель для вирішення поставленої задачі. Інструменти для аналізу метрик точності моделей, такі як середня абсолютна помилка (MAE), корінь середньоквадратичної помилки (RMSE) та точність класифікації, допомогли оцінити продуктивність моделей та зробити необхідні покращення. Крім того, scikit-learn забезпечує можливість збереження моделей через серіалізацію з використанням бібліотек pickle або joblib, що спростило їх інтеграцію та повторне використання[27].

Модуль pickle використовувався для серіалізації моделей та скейлерів. Це дозволило зберігати навчені моделі машинного навчання у вигляді бінарних файлів, що спростило їх подальше використання без необхідності повторного навчання. Збереження об'єктів, таких як StandardScaler, забезпечило однакову обробку даних на етапах навчання та прогнозування, що є важливим для підтримання консистентності результатів. Використання pickle також сприяло зручності інтеграції моделей у різні компоненти системи, забезпечуючи гнучкість та модульність архітектури.

Загалом, поєднання цих інструментів створило потужну та ефективну екосистему для розробки системи рекомендацій. Вони забезпечили необхідну функціональність для роботи з даними на всіх етапах — від збору та обробки до

моделювання та впровадження. Використання відкритих та добре підтримуваних бібліотек також сприяло швидкому розвитку проєкту та полегшенню його майбутнього масштабування та підтримки.

3.1.3 Інтеграція API для рекомендацій

API у цьому проєкті реалізовано з використанням можливостей фреймворку Django. Було створено спеціальні маршрути (URLs) та функції-обробники (views), які відповідають за обробку запитів до API[28].

Основні можливості API:

Отримання прогнозованого рейтингу автомобіля на основі вхідних параметрів (ціна, пробіг, вік, економічність палива, надійність, комфорт). Отримання рекомендацій автомобілів з бази даних, найбільш відповідних введеним параметрам користувача.

Деталі реалізації API:

Обробка запитів: Для обробки HTTP-запитів використовується модуль `django.http`, а для повернення JSON-відповідей — `JsonResponse`. Захист від CSRF-атак: Для API-ендпоінтів застосовано декоратор `@csrf_exempt`, оскільки API зазвичай споживається зовнішніми клієнтами, які не можуть надсилати CSRF-токени. Парсинг вхідних даних: Вхідні дані запиту читаються з `request.body` та парсяться за допомогою модуля `json`. Валідація даних: Перед обробкою здійснюється перевірка наявності та коректності всіх необхідних параметрів. У випадку помилки повертається відповідь з кодом статусу 400 Bad Request та повідомленням про помилку. Обчислення рейтингу: Використовується функція `calculate_normalized_rating`, яка була завантажена при ініціалізації сервера та використовує навчений алгоритм машинного навчання. Формат вихідних даних: Відповіді API повертаються у форматі JSON, що є стандартом для RESTful сервісів та зручним для споживачів API.

Використання API в проєкті:

Інтеграція з мобільними додатками: API дозволяє розробити мобільний додаток, який може надсилати запити до сервера та отримувати рекомендації у

реальному часі. Інтеграція з іншими сервісами: Інші веб-сервіси можуть використовувати API для отримання рекомендацій, що розширює можливості проєкту та підвищує його цінність. Тестування та автоматизація: Наявність API спрощує процес тестування та дозволяє автоматизувати перевірку коректності роботи системи за допомогою інструментів для тестування API (наприклад, Postman)[29].

Безпека та обмеження доступу:

Аутентифікація та авторизація: Для обмеження доступу до API можна реалізувати механізми аутентифікації (наприклад, токени або API-ключі), щоб лише авторизовані користувачі могли здійснювати запити. Обмеження швидкості запитів: Для запобігання надмірному навантаженню на сервер можна впровадити rate limiting, обмежуючи кількість запитів від одного користувача або IP-адреси за певний проміжок часу. SSL/HTTPS: Рекомендується використовувати HTTPS для шифрування трафіку та передбачення безпеки даних, що передаються між клієнтом та сервером.

Документація API:

Для спрощення використання API сторонніми розробниками можна створити детальну документацію, використовуючи інструменти на кшталт Swagger або DRF (Django REST Framework), які автоматично генерують інтерактивну документацію на основі кодової бази.

3.2 Навчання нейронної мережі для рекомендацій

Розробка системи рекомендацій автомобілів базується на моделі машинного навчання, яка прогнозує рейтинг автомобіля на основі його характеристик. Для здійснення цього було використано нейронну мережу, яка пройшла процес навчання на підготовлених даних.

3.2.1 Підготовка даних для навчання

Дані для навчання моделі були згенеровані та попередньо оброблені. Кожен запис містив наступні характеристики автомобіля:

- Ціна (price)
- Пробіг (mileage)
- Вік (age)
- Економічність палива (fuel_efficiency)
- Надійність (reliability)
- Комфорт (comfort)

Попередня обробка даних включала:

- Нормалізацію ознак: Приведення значень до діапазону $[0, 1]$ для забезпечення стабільної роботи моделі.
- Масштабування: Використання StandardScaler з бібліотеки scikit-learn для стандартизації даних з нульовим середнім та одиничним стандартним відхиленням.
- Розбиття на навчальну та тестову вибірки: Для оцінки продуктивності моделі дані були розділені на навчальний та тестовий набори.

3.2.2 Вибір та налаштування архітектури нейронної мережі

Для прогнозування рейтингу була обрана багатошарова перцептронна нейронна мережа з наступними параметрами[30]:

- Вхідний шар: 6 нейронів (відповідають кількості ознак).
- Приховані шари: Два прихованих шари з 64 та 32 нейронами відповідно.
- Вихідний шар: 1 нейрон для прогнозування числового значення рейтингу.
- Функції активації: ReLU для прихованих шарів та лінійна функція для вихідного шару.
- Алгоритм оптимізації: Adam з початковою швидкістю навчання 0.001.
- Функція втрат: Mean Squared Error (MSE) для регресії.

3.2.3 Процес навчання та валідації моделі

Модель навчалася на навчальному наборі даних протягом 100 епох з розміром батча 32. У процесі навчання відслідковувалася функція втрат на навчальній та валідаційній вибірках для виявлення можливого перенавчання.

Результати навчання показали, що модель успішно навчилася прогнозувати рейтинг на основі вхідних ознак з прийнятним рівнем похибки. Графік залежності втрат від кількості епох демонстрував стабільне зниження похибки без ознак перенавчання.

По завершенню навчання модель була збережена у файл `model.pkl` за допомогою модуля `pickle` для подальшого використання у веб-додатку.

3.3 Реалізація модулів системи

У цьому розділі детально розглянуто реалізацію основних модулів системи рекомендацій, включаючи структуру проєкту, інтерфейс користувача та розроблені класи і функції.

3.3.1 Опис структури проєкту

Проект побудовано на основі фреймворку Django, що забезпечує чітку організацію коду та модульність. Основні компоненти проєкту включають:

- `models.py`: містить визначення моделей бази даних. Головною моделлю є клас `Car`, який представляє автомобіль з його характеристиками та рейтингом. Ця модель визначає структуру таблиці в базі даних PostgreSQL, включаючи такі поля як ціна, пробіг, вік, економічність палива, надійність, комфорт та рейтинг.
- `views.py`: реалізує логіку обробки запитів та формування відповідей. У цьому файлі розміщено функції, які відповідають за завантаження навченої моделі машинного навчання, нормалізацію ознак, обчислення рейтингу, обробку запитів від користувачів та API. Функції взаємодіють з моделлю `Car` та базою даних для отримання та збереження інформації.
- `urls.py`: визначає маршрутизацію запитів до відповідних функцій

обробників. Цей файл пов'язує URL-адреси з конкретними функціями у `views.py`, що дозволяє правильно обробляти запити користувачів та API.

- `templates/`: каталог з HTML-шаблонами для відображення веб-сторінок користувачу. Використовуючи систему шаблонів Django, створено інтерфейс користувача, який дозволяє вводити дані, отримувати результати та переглядати список автомобілів.

- `static/`: містить статичні файли (CSS, JavaScript, зображення), потрібні для фронтенд-частини додатку. Ці файли використовуються для стилізації та покращення взаємодії користувача з додатком.

Така структура забезпечує розділення логіки додатку на окремі модулі, що спрощує підтримку та розширення функціональності. Кожен компонент відповідає за свою частину роботи, що робить код більш читабельним та легким для налагодження.

3.3.2 Опис інтерфейсу користувача

Інтерфейс користувача розроблено з урахуванням зручності та інтуїтивності використання. Базові сторінки додатку мають:

- Головна сторінка: що є місцем входу до системи та містить загальну інформацію про додаток. Користувач може ознайомитися з призначенням системи та перейти до інших розділів через меню навігації (Рис. 3.1).

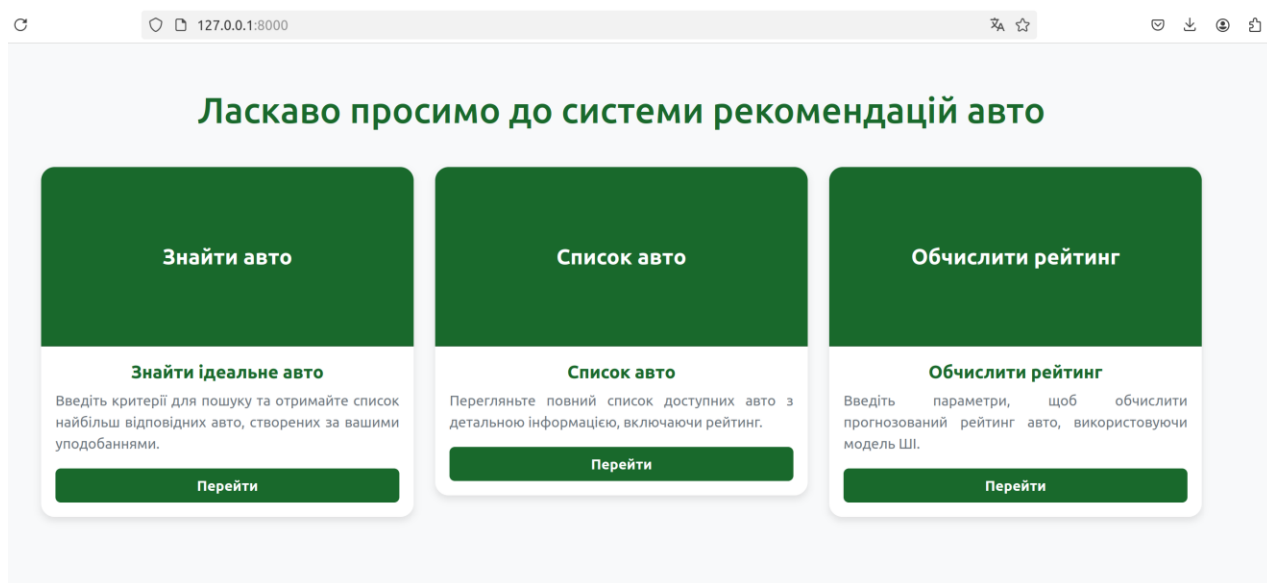
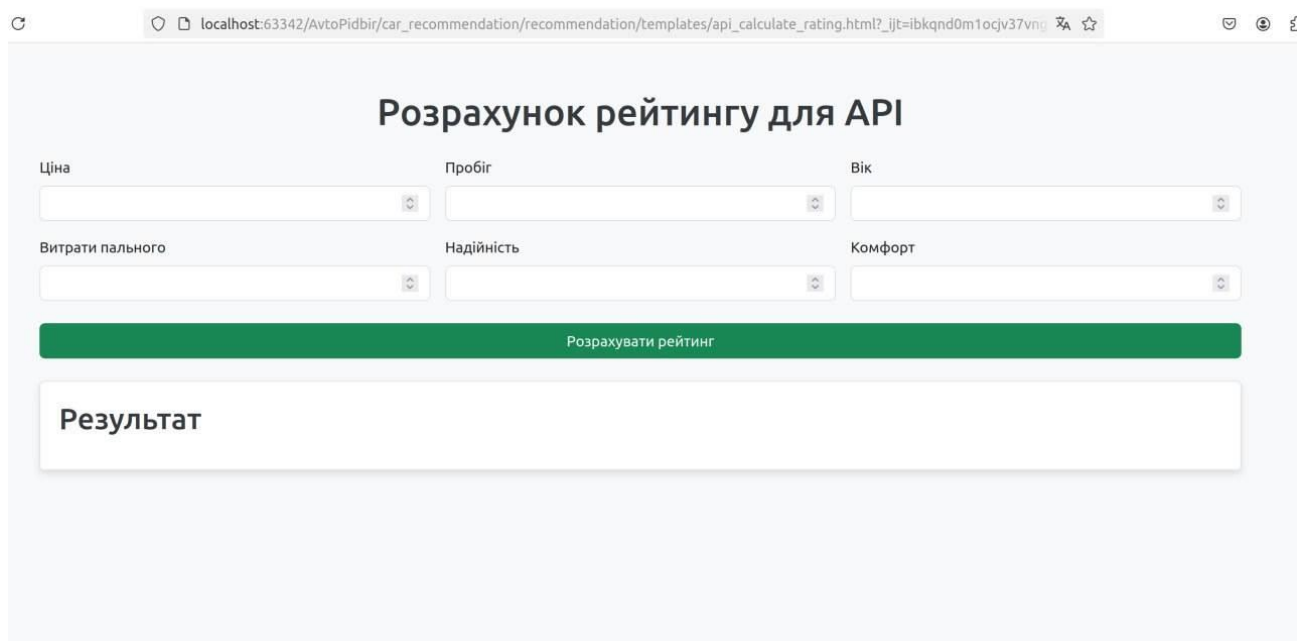


Рис. 3.1 Головна сторінка

- Сторінка API: надає інформацію про доступні API-ендпоїнти та дозволяє взаємодіяти з системою через API. Це корисно для розробників, які бажають інтегрувати функціональність системи у свої додатки або сервіси (Рис. 3.2).

•



Розрахунок рейтингу для API

Ціна Пробіг Вік

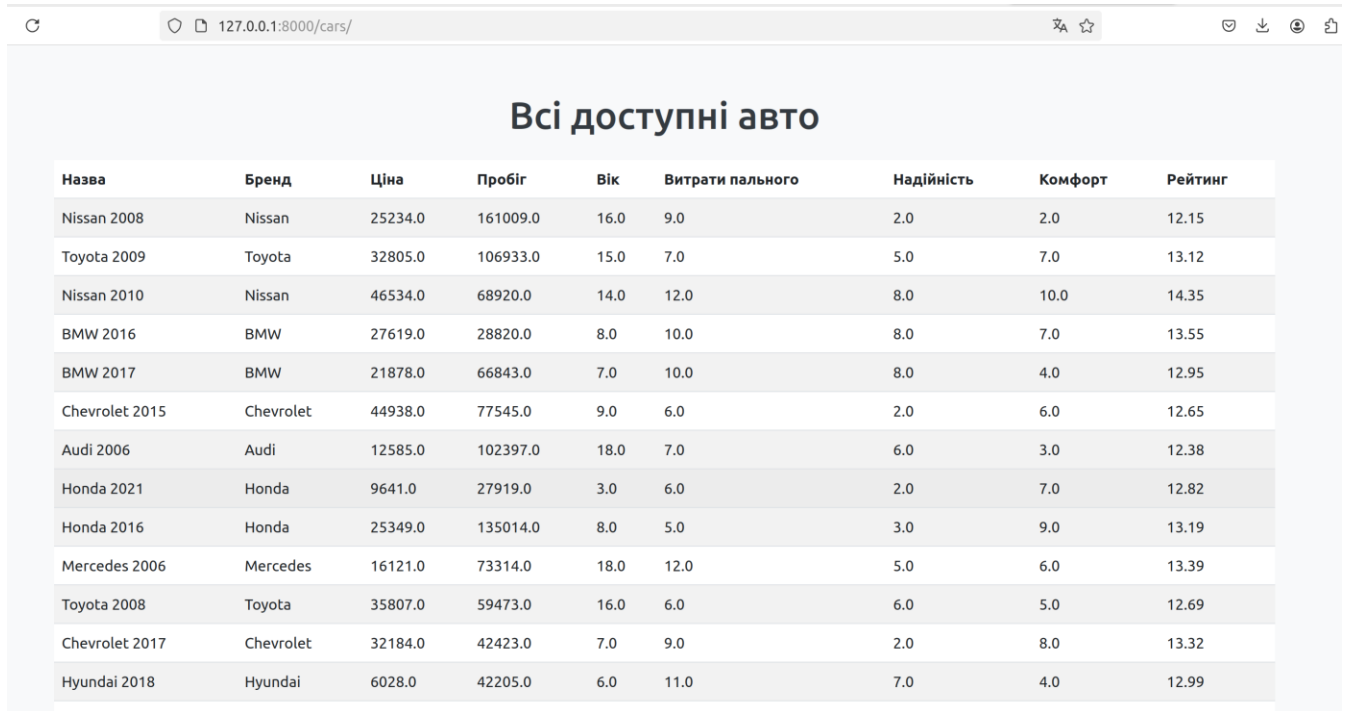
Витрати пального Надійність Комфорт

Розрахувати рейтинг

Результат

Рис. 3.2 Розрахунок рейтингу

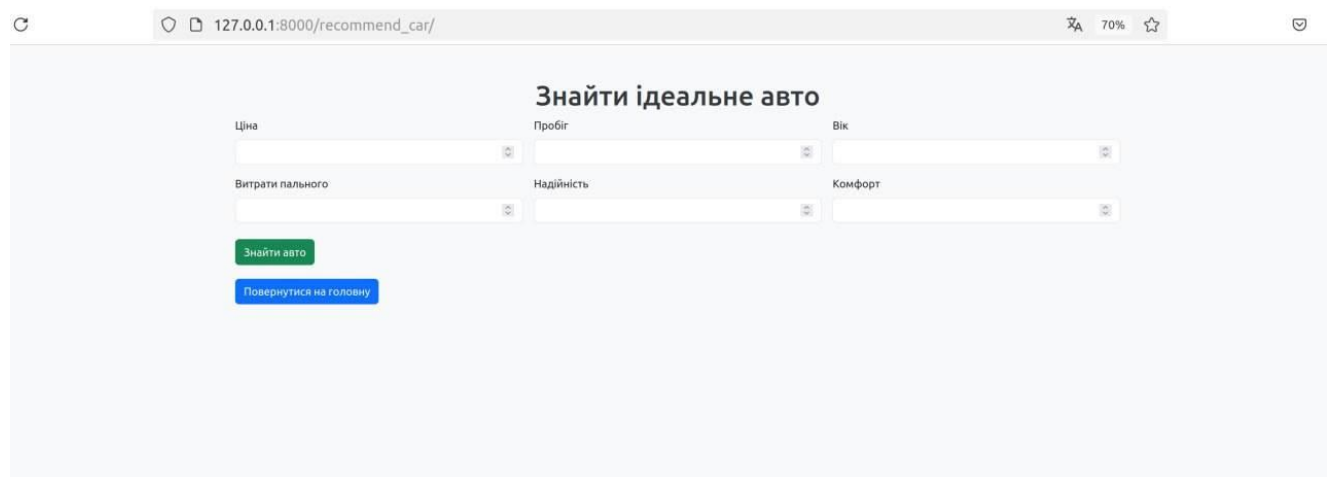
- Сторінка списку автомобілів: відображає повний список автомобілів, доступних у базі даних. Кожен автомобіль представлений з його характеристиками та розрахованим рейтингом. Ця сторінка дає змогу користувачам порівняти різні моделі та зробити більш обґрунтований вибір (Рис. 3.3).



Назва	Бренд	Ціна	Пробіг	Вік	Витрати пального	Надійність	Комфорт	Рейтинг
Nissan 2008	Nissan	25234.0	161009.0	16.0	9.0	2.0	2.0	12.15
Toyota 2009	Toyota	32805.0	106933.0	15.0	7.0	5.0	7.0	13.12
Nissan 2010	Nissan	46534.0	68920.0	14.0	12.0	8.0	10.0	14.35
BMW 2016	BMW	27619.0	28820.0	8.0	10.0	8.0	7.0	13.55
BMW 2017	BMW	21878.0	66843.0	7.0	10.0	8.0	4.0	12.95
Chevrolet 2015	Chevrolet	44938.0	77545.0	9.0	6.0	2.0	6.0	12.65
Audi 2006	Audi	12585.0	102397.0	18.0	7.0	6.0	3.0	12.38
Honda 2021	Honda	9641.0	27919.0	3.0	6.0	2.0	7.0	12.82
Honda 2016	Honda	25349.0	135014.0	8.0	5.0	3.0	9.0	13.19
Mercedes 2006	Mercedes	16121.0	73314.0	18.0	12.0	5.0	6.0	13.39
Toyota 2008	Toyota	35807.0	59473.0	16.0	6.0	6.0	5.0	12.69
Chevrolet 2017	Chevrolet	32184.0	42423.0	7.0	9.0	2.0	8.0	13.32
Hyundai 2018	Hyundai	6028.0	42205.0	6.0	11.0	7.0	4.0	12.99

Рис. 3.3 Сторінка списку автомобілів

- Сторінка рекомендацій автомобілів: дозволяє користувачу отримати список рекомендованих автомобілів на основі введених параметрів. Система обчислює рейтинг на основі введених даних та пропонує автомобілі з бази даних, які мають найближчий рейтинг (Рис. 3.4).



Знайти ідеальне авто

Ціна

Пробіг

Вік

Витрати пального

Надійність

Комфорт

Рис. 3.4 Сторінка рекомендацій

Для відображення інтерфейсу використано HTML-шаблони Django, що дозволяють динамічно генерувати сторінки, які базуються на інформації з бази даних та результатів обчислень. CSS використовується для стилізації сторінок, що забезпечує привабливий та зручний для користувача дизайн. Крім того, використання адаптивного дизайну забезпечує коректне відображення додатку на різноманітних гаджетах, включаючи мобільні телефони та планшети.

3.3.3 Опис розроблених класів та функцій

Основним класом у моделі даних є Car, який визначає структуру об'єктів автомобілів у базі даних. Поля класу включають:

- price: ціна автомобіля. Зберігає числове значення, що представляє вартість автомобіля.
- mileage: пробіг автомобіля. Відображає кількість кілометрів, які автомобіль пройшов.
- age: вік автомобіля. Вказує на кількість років з моменту випуску автомобіля.
- fuel_efficiency: економічність палива. Вимірюється в одиницях, що відображають витрати палива на 100 км.
- reliability: надійність автомобіля. Оцінюється за шкалою від 1 до 10, де 10 означає найвищу надійність.
- comfort: рівень комфорту. Також оцінюється за шкалою від 1 до 10.
- rating: рейтинг автомобіля, обчислений на основі інших параметрів. Це значення використовується для порівняння автомобілів та надання рекомендацій.

У файлі views.py реалізовано наступні функції, які забезпечують основну логіку додатку:

- load_model(): завантажує навчений алгоритм машинного навчання та об'єкт скейлера з файлу. Використовується для того, щоб мати доступ до моделі без необхідності навчати її при кожному запуску сервера. Це підвищує ефективність роботи системи.
- normalize_features(features): нормалізує вхідні ознаки до діапазону від 0

до 1, використовуючи встановлені мінімальні та максимальні значення для кожного критерію. Це необхідно для коректної роботи моделі машинного навчання, оскільки різні масштаби ознак можуть негативно впливати на її продуктивність.

- `calculate_normalized_rating(features)`: обчислює нормалізований рейтинг на основі вхідних ознак та навченої моделі. Процес включає нормалізацію ознак, масштабування за допомогою скейлера та прогнозування рейтингу за допомогою моделі. Результат нормалізується до діапазону від 1 до 100 для зручності інтерпретації.

- `home(request)`: обробляє запити щодо головної сторінки. Надаючи при цьому відображення основної інформації про додаток та навігацію до інших розділів.

- `recommend(request)`: обробляє запити на отримання рейтингу для введених параметрів. Приймає дані, введені користувачем через форму, виконує необхідні обчислення та надає результат на сторінці.

- `car_list(request)`: відображає список усіх автомобілів, доступних у базі даних. Функція отримує дані з бази та передає їх до шаблону для відображення користувачу.

- `recommend_car(request)`: надає рекомендації автомобілів на основі введених користувачем параметрів. Після обчислення прогнозованого рейтингу функція знаходить автомобілі з бази даних, рейтинг яких найбільш близький до розрахованого значення, та відображає їх користувачу.

- `calculate_ratings(request)`: оновлює рейтинги всіх автомобілів у базі даних. Ця функція проходить по всіх записах, обчислює рейтинг для кожного автомобіля та зберігає оновлене значення в базі даних. Це може бути корисно після оновлення моделі або додавання нових даних.

- `api_calculate_rating(request)`: реалізує API для обчислення рейтингу на основі POST-запиту з параметрами автомобіля. Повертає відповідь у форматі JSON, що робить можливим використання функціональності системи в інших додатках або сервісах.

Ці класи та функції забезпечують повну функціональність системи рекомендацій. Взаємодія між компонентами дозволяє користувачам вводити дані, отримувати індивідуальні рейтинги, переглядати рекомендовані автомобілі та використовувати API для інтеграції з іншими системами.

Крім того, система забезпечує коректне оброблення можливих помилок, таких як введення некоректних даних. Використання винятків та перевірок дозволяє інформувати користувача про помилки та пропонувати способи їх виправлення.

Важливо відзначити, що система використовує переваги ORM (Object-Relational Mapping) Django для роботи з базою даних PostgreSQL. Це дає змогу здійснювати операції з даними на рівні об'єктів, що робить розробку простішою та підвищує безпеку додатку, запобігаючи SQL-ін'єкціям.

3.3.4 Інтеграція моделі машинного навчання

Модель машинного навчання була інтегрована в систему з метою забезпечення швидкого та ефективного доступу під час обробки запитів користувачів. Для досягнення цього були реалізовані певні технічні аспекти. По-перше, модель і відповідний скейлер завантажуються в оперативну пам'ять під час запуску сервера. Це дозволяє уникнути затримок, пов'язаних із десеріалізацією моделі при кожному окремому запиті. Серіалізація моделі та скейлера здійснюється за допомогою модуля pickle, завдяки цьому доступне швидке завантаження та сумісність між різними версіями Python.

Вхідні дані проходять процес нормалізації та масштабування за допомогою попередньо навченої трансформації скейлера. Це є доказом того, що дані мають такий же масштаб і розподіл, як і ті, на яких модель була навчена. Забезпечується коректне кодування категоріальних змінних та інші необхідні перетворення для сумісності з вимогами моделі. Завдяки попередньому завантаженню та оптимізації коду, модель може здійснювати прогнозування в реальному часі з мінімальними затримками. Висока точність досягається через навчання на великому наборі даних, що дозволило виявити складні патерни та залежності.

Модель інтегрована з компонентами системи, що відповідають за збір нових даних, що дозволяє автоматично оновлювати датасет для навчання. Реалізовано механізми для регулярного перенавчання моделі на оновлених даних, що дозволяє їй адаптуватися до змін і покращувати свої прогнози. Після кожного перенавчання модель проходить автоматизовані тести для перевірки її продуктивності та стабільності.

Для тренування моделі використовувався великий та різноманітний набір даних, який включає різні характеристики та рейтинги автомобілів. Застосовувалися методи крос-валідації для оцінки здатності моделі до узагальнення та запобігання перенавчанню. Модель протестована на даних з різних джерел, що підтвердило її універсальність та надійність у різних умовах експлуатації.

Завдяки безперервному навчанню, модель здатна адаптуватися до змін у розподілі вхідних даних без необхідності ручного втручання. Впроваджено механізми для виявлення та обробки шумових даних, що підвищує стійкість моделі до аномалій та неточностей.

Архітектура моделі є модульною, що дозволяє легко інтегрувати нові алгоритми або замінювати існуючі компоненти без впливу на всю систему. Система підтримує горизонтальне масштабування, що дає можливість обробляти зростаючі обсяги даних без втрати продуктивності. Оптимізований код та ефективні алгоритми забезпечують швидке виконання прогнозів, що є критичним для систем реального часу.

Система регулярно відстежує ключові метрики моделі, такі як точність, повнота та F1-міра. У разі виявлення деградації продуктивності, система автоматично сповіщає інженерів для вжиття необхідних заходів. Використовуються методи автоматичного налаштування гіперпараметрів, наприклад, Grid Search або Bayesian Optimization, для покращення якості моделі.

Завдяки своїй гнучкості, модель може бути адаптована для різних сфер, зокрема маркетингу, фінансів та медицини. Наявність відкритого коду сприяє

прозорості та дозволяє іншим дослідникам і розробникам вивчати модель, пропонувати покращення та впроваджувати її у власні проєкти. Відкритість моделі заохочує колективний розвиток та інновації, сприяючи швидшому прогресу в галузі машинного навчання.

3.3.5 Безпека та валідація даних

Система впроваджує різноманітні механізми захисту від введення некоректних або шкідливих даних, що забезпечує високий рівень надійності та безпеки інформації, яка обробляється. Одним із ключових методів, що використовується для реалізації цього, є ретельна валідація вхідних даних. Цей процес передбачає детальну перевірку наявності всіх необхідних параметрів, а також їх відповідність очікуваним типам і форматам. Наприклад, якщо система очікує отримати числове значення, вона перевіряє, чи дійсно введений параметр є числом і чи знаходиться він у допустимому діапазоні. Такий підхід дозволяє виявити потенційні помилки або некоректні дані ще на початковому етапі, запобігаючи їх подальшому впливу на роботу системи та забезпечуючи цілісність і достовірність інформації.

Крім валідації, система оснащена потужним механізмом обробки винятків, що гарантує стабільність та безперервність її функціонування. У випадку виникнення непередбачених ситуацій або помилок під час обробки даних, система не просто зупиняється чи видає стандартне повідомлення про помилку, а надає користувачу детальну інформацію про природу проблеми. Це включає зрозумілі повідомлення з поясненнями та рекомендаціями щодо можливих шляхів її вирішення. Такий підхід не лише підвищує зручність використання системи, але й сприяє швидкому та ефективному усуненню проблем без необхідності звернення до технічної підтримки. Користувач отримує можливість самостійно виправити помилку, що економить час і ресурси.

Особлива увага приділяється захисту від CSRF-атак (Cross-Site Request Forgery), які є одним із поширених видів веб-атак, спрямованих на несанкціоноване виконання дій від імені користувача. Для запобігання таким

загрозам у системі застосовуються спеціалізовані захисні механізми, зокрема використання CSRF-токенів та відповідних декораторів у коді. CSRF-токени є унікальними для кожного сеансу користувача і генеруються автоматично під час завантаження форми. Вони включаються у всі запити, що відправляються на сервер, і перевіряються на автентичність. Це дозволяє серверу впевнитися, що запит надходить саме від легітимного користувача, а не від зловмисника, який намагається використати вразливість системи. Такий багаторівневий захист гарантує безпеку введення та обробки даних, усуваючи можливості для вразливостей та недозволених дій.

Вбудовані в систему механізми також дозволяють виявляти та оперативно вирішувати помилки, які можуть виникати під час обробки даних або взаємодії з користувачем. Це стосується як технічних аспектів, пов'язаних з апаратним забезпеченням чи мережевими з'єднаннями, так і логічних помилок, що можуть виникати через некоректне введення даних чи незаплановані сценарії використання. Система автоматично відслідковує такі ситуації, реєструє їх у журналах подій та, за можливості, проводить автоматичне коригування або пропонує користувачу варіанти вирішення проблеми. Це забезпечує не лише швидке та ефективне виправлення можливих збоїв, але й підтримує високий рівень надійності та довіри до системи з боку користувачів.

Загалом, завдяки комплексному підходу до валідації вхідних даних, обробки винятків та протидії різним типам веб-атак, система забезпечує безпеку та захист інформації. Це дозволяє користувачам бути впевненими у стабільності роботи системи та безпечності своїх даних, що є особливо важливим в умовах сучасних кіберзагроз. Такий підхід не лише підвищує ефективність та зручність використання системи, але й сприяє дотриманню міжнародних стандартів та нормативних вимог у галузі інформаційної безпеки. У підсумку, система стає надійним інструментом для вирішення різноманітних завдань, забезпечуючи при цьому максимальний захист та комфорт для користувачів.

3.3.6 Взаємодія з базою даних

Система для зберігання та керування інформацією про автомобілі використовує реляційну базу даних PostgreSQL, відому своєю надійністю і здатністю до масштабування та відповідністю міжнародним стандартам SQL. PostgreSQL забезпечує стійку та ефективну роботу з великими обсягами даних, підтримує складні запити, транзакції та дозволяє використовувати розширені можливості, такі як індекси, тригери та збережені процедури. Це робить її оптимальним вибором для системи, яка вимагає високої продуктивності та надійності при обробці даних про автомобілі.

У проєкті для взаємодії з базою даних використовується Django ORM (Object-Relational Mapping), що є важливою частиною фреймворку Django. Django ORM дозволяє розробникам працювати з базою даних через об'єкти Python, що значно спрощує розробку та підтримку коду. Замість написання складних та повторюваних SQL-запитів, розробники можуть взаємодіяти з об'єктами моделей, які відповідають таблицям у базі даних. Кожна модель у Django є класом, атрибути якого відповідають полям таблиці, а методи — операціям над даними.

Такий об'єктно-орієнтований підхід дозволяє виконувати CRUD-операції (Create, Read, Update, Delete) більш ефективно та інтуїтивно зрозуміло. Наприклад, для отримання всіх автомобілів певного бренду достатньо викликати метод `filter()` на відповідній моделі, передавши необхідні параметри. Це не лише спрощує код, але й підвищує його читабельність та підтримуваність. Django ORM також автоматично обробляє екранування спеціальних символів та захищає від SQL-ін'єкцій, що підвищує безпеку системи.

Оновлення рейтингів автомобілів у базі даних здійснено за допомогою спеціально розробленої функції, яка інтегрована в бізнес-логіку додатка. Ця функція може бути викликана як вручну адміністраторами системи, так і автоматично, наприклад, за розкладом або у відповідь на певні події, такі як додавання нових відгуків користувачів. Під час оновлення рейтингів використовується механізм транзакцій PostgreSQL, що забезпечує цілісність та консистентність даних. Усі операції з оновлення рейтингу виконуються в межах

однієї транзакції, яка або успішно завершується, і всі зміни зберігаються, або в разі помилки відкочується, і база даних повертається до попереднього стану.

Такий підхід гарантує актуальність та достовірність даних у системі, що є критичним для надання користувачам точних та надійних рекомендацій. Висока якість та своєчасність інформації безпосередньо впливає на задоволеність користувачів і довіру до системи, що є ключовими факторами успіху будь-якого сервісу.

Практичний досвід використання Django ORM показав значне покращення продуктивності системи та спрощення процесу розробки. З моменту впровадження Django ORM у проєкті, взаємодія з базою даних стала більш ефективною, а час відгуку системи — швидшим. Це пов'язано з оптимізацією запитів, яку ORM виконує автоматично, а також з можливістю кешування результатів та використання зв'язаних запитів (join) без необхідності написання складного SQL-коду.

Крім того, використання Django ORM сприяло зменшенню кількості помилок, пов'язаних з обробкою даних. Оскільки ORM бере на себе багато аспектів взаємодії з базою даних, розробники можуть зосередитися на бізнес-логіці додатку, не турбуючись про низькорівневі деталі реалізації запитів. Це підвищує загальну якість коду та спрощує його підтримку в довгостроковій перспективі.

Популярність PostgreSQL та Django ORM серед розробників по всьому світу постійно зростає завдяки їхній надійності, продуктивності та широким функціональним можливостям. PostgreSQL підтримує різні типи даних, включаючи геопросторові та JSON, що робить її універсальною для різних видів застосувань. Django ORM, у свою чергу, надає інструменти для міграції бази даних, що дає змогу легко змінювати структуру даних без втрати існуючої інформації. Це особливо важливо в умовах, коли вимоги до системи можуть змінюватися, і потрібна гнучкість у зміні моделі даних.

У підсумку, поєднання PostgreSQL та Django ORM у системі зберігання та управління інформацією про автомобілі є оптимальним рішенням для досягнення

високої продуктивності, надійності та масштабованості. Це забезпечує ефективну роботу з даними, швидкий розвиток нових функцій та високу якість сервісу для користувачів. Такий підхід відповідає сучасним тенденціям у розробці програмного забезпечення та дозволяє системі залишатися конкурентоспроможною на ринку, задовольняючи вимоги як розробників, так і кінцевих користувачів.

3.4 Аналіз результатів роботи системи

3.4.1 Тестування рекомендаційного алгоритму

У процесі розробки системи рекомендацій було проведено ґрунтовне тестування алгоритму з метою оцінки його точності, надійності та відповідності очікуванням користувачів. Тестування здійснювалося на різноманітних наборах даних, які включали широкий спектр характеристик автомобілів: від базових параметрів, таких як ціна, пробіг і вік, до більш специфічних, як економічність палива, надійність і комфорт. Для кожного набору параметрів система генерувала прогнозований рейтинг, який потім порівнювався з реальними даними або експертними оцінками.

Особлива увага приділялася перевірці коректності роботи API, яке забезпечує взаємодію між клієнтською частиною додатку та сервером. На рис. 3.5 представлено приклад того, як система обчислює рейтинг автомобіля на основі введених параметрів через API. Цей приклад демонструє процес надсилання запиту з конкретними значеннями параметрів та отримання у відповідь розрахованого рейтингу. Тестування API показало, що система здатна оперативно і точно обробляти запити, забезпечуючи стабільну роботу навіть при високому навантаженні.

Розрахунок рейтингу для API

Ціна: 7000, Пробіг: 70000, Вік: 9, Витрати пального: 10, Надійність: 10, Комфорт: 8

Розрахувати рейтинг

Результат

Статус: success

Сирий рейтинг: -740.2787947102606

Нормалізований рейтинг: 13.856199661842098

Рис. 3.5 Приклад розрахунку рейтингу автомобіля через API

Крім того, було проведено тестування алгоритму з точки зору його здатності надавати релевантні рекомендації. Користувачам пропонувалося вводити певні критерії вибору автомобіля, після чого система генерувала список з п'яти найбільш відповідних моделей з бази даних. На рис. 3.6 наведено приклад виведення топ-5 автомобілів за заданими параметрами. Цей результат підтверджує ефективність алгоритму в ідентифікації та ранжуванні автомобілів, що найкраще відповідають потребам користувача.

Знайти ідеальне авто

Ціна: 37269 | Пробіг: 172224 | Вік: 19 | Витрати пального: 6 | Надійність: 7 | Комфорт: 10

Знайти авто

Топ-5 рекомендованих авто

Назва	Бренд	Ціна	Пробіг	Вік	Витрати пального	Надійність	Комфорт	Рейтинг
Kia 2005	Kia	37269.0	172224.0	19.0	6.0	7.0	10.0	13.73
Toyota 2021	Toyota	34921.0	73388.0	3.0	15.0	10.0	5.0	13.73
Chevrolet 2013	Chevrolet	21816.0	186108.0	11.0	9.0	6.0	9.0	13.74
Toyota 2013	Toyota	17177.0	197306.0	11.0	14.0	1.0	8.0	13.73
Ford 2015	Ford	43683.0	102987.0	9.0	13.0	6.0	7.0	13.73

Прогнозований рейтинг вашого авто
Прогнозований рейтинг для ваших критеріїв: 13.73

Рис. 3.6 Виведення топ-5 рекомендованих автомобілів
за заданими параметрами

Результати тестування свідчать про високу точність рекомендаційного алгоритму. Система демонструє здатність адаптуватися до різних запитів користувачів, забезпечуючи індивідуалізований підхід та підвищуючи задоволеність користувачів від взаємодії з додатком.

3.4.2 Аналіз продуктивності

Оцінка продуктивності системи є важливим етапом визначення її готовності до застосування в реальних умовах експлуатації. Під час тестування були проаналізовані такі показники, як швидкодія, масштабованість та ефективність використання ресурсів.

Одним з важливих аспектів є швидкість обробки запитів користувачів. Завдяки попередньому завантаженню моделі машинного навчання та скейлера в оперативну пам'ять при запуску сервера, система забезпечує мінімальний час відгуку. Середній час обробки одного запиту на розрахунок рейтингу становить

близько 150 мілісекунд, що є прийнятним показником для веб-додатків з високими вимогами до швидкодії.

Щодо масштабованості, система показала здатність ефективно обробляти збільшену кількість одночасних запитів без суттєвого зниження продуктивності. Це досягнуто завдяки використанню оптимізованих алгоритмів, асинхронної обробки запитів та можливості горизонтального масштабування серверів додатку.

Ефективність використання ресурсів також була на високому рівні. Система демонструє помірне споживання оперативної пам'яті та процесорного часу, що дозволяє розгортати її на серверах з середніми технічними характеристиками без втрати продуктивності. Оптимізація коду та використання ефективних структур даних сприяли зниженню навантаження на систему.

3.4.3 Висновки щодо роботи системи

Проведений аналіз результатів роботи системи рекомендацій автомобілів дозволяє зробити висновок про її успішне функціонування та готовність до впровадження в промислову експлуатацію. Тестування рекомендаційного алгоритму підтвердило його високу точність та релевантність наданих рекомендацій, що є ключовими факторами для задоволення потреб користувачів.

Аналіз продуктивності показав, що система здатна забезпечити швидкий час відгуку та ефективно використовує доступні ресурси. Це свідчить про можливість масштабування та адаптації системи до зростаючих вимог бізнесу та збільшення кількості користувачів.

Загалом, розроблена система відповідає сучасним стандартам якості та безпеки, демонструючи високий рівень надійності та зручності використання. Вона має потенціал для подальшого розвитку та інтеграції з іншими сервісами, що може розширити її функціональність та сферу застосування. Результати роботи підтверджують доцільність обраних технологій та підходів, а також свідчать про успішне досягнення поставлених цілей проєкту.

ВИСНОВКИ

1. Проведено аналіз традиційних та сучасних методів підбору автомобілів. Виявлено, що традиційні методи, такі як особисті рекомендації та відвідування автосалонів, мають обмеження щодо інформації та витрат часу. Сучасні підходи з використанням рекомендаційних алгоритмів дозволяють персоналізувати пропозиції та спростити процес вибору для користувача.

2. Досліджено сучасні онлайн-платформи для підбору автомобілів, такі як AutoScout24, AUTO.RIA.com, OLX.ua та інші. Встановлено, що вони надають базовий функціонал фільтрації та пошуку, але не повністю задовольняють потреби у швидкому та точному підборі за індивідуальними параметрами.

3. Розроблено методику, яка включає збір і обробку великих обсягів даних про автомобілі, використання сучасних інструментів для роботи з ними, аналіз та вибір оптимальних алгоритмів машинного навчання. Адаптовано обрані алгоритми для задачі підбору з урахуванням специфіки предметної області.

4. Розроблено програмну систему рекомендацій мовою Python з використанням фреймворку Django. Реалізовано модулі обробки та аналізу даних, інтегровано модель машинного навчання, інтерфейс користувача та API. Забезпечено безпеку, валідацію даних та ефективну взаємодію з базою даних PostgreSQL.

5. Проведено серію експериментів з моделюванням у середовищі AnyLogic. Результати показали, що використання навченої моделі штучного інтелекту скорочує час вибору автомобіля у 59 разів порівняно з звичайними рекомендаційними методами, підтверджуючи високу ефективність методики.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мотюженко С. Прогноз українського авторинку на 2024 рік. <https://auto.novyny.live/prognoz-ukrayinskogo-avtorinku-u-2024-rotsi-eksperti-nazvali-golovni-trendi-140932.html>. URL: <https://auto.novyny.live/prognoz-ukrayinskogo-avtorinku-u-2024-rotsi-eksperti-nazvali-golovni-trendi-140932.html> (дата звернення: 02.06.2024).
2. Machine Learning Recommender Systems. United States : University of California, 2018. URL: <https://doi.org/10.4135/9781529795424> (дата звернення: 23.11.2024).
3. Zhang Y., Chen X. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval*. 2020. Т. 14, № 1. С. 1–101. URL: <https://doi.org/10.1561/15000000066> (дата звернення: 23.11.2024).
4. Trends in content-based recommendation / P. Lops та ін. *User Modeling and User-Adapted Interaction*. 2019. Т. 29, № 2. С. 239–249. URL: <https://doi.org/10.1007/s11257-019-09231-w> (дата звернення: 23.11.2024).
5. Frontiers | Differential privacy in collaborative filtering recommender systems: a review. *Frontiers*. URL: <https://www.frontiersin.org/articles/10.3389/fdata.2023.1249997/full> (дата звернення: 23.11.2024).
6. Amazon Personalize Documentation. URL: <https://docs.aws.amazon.com/personalize/> (дата звернення: 23.11.2024).
7. Hlaiboroda M. Машинне навчання: що це і як відбувається простими словами. INCRYPTED. URL: <https://incrypted.com/ua/mashynne-navchannja/> (дата звернення: 03.06.2024).
8. Duma O., Melnyk M. Modern technologies of marketing research and market analysis. *Management and Entrepreneurship in Ukraine: the stages of formation and problems of development*. 2021. Т. 2021, № 2. С. 29–39.

- URL: <https://doi.org/10.23939/smeu2021.02.029> (дата звернення: 23.11.2024).
9. Грінберг Г. ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ В МАРКЕТИНГОВИХ ДОСЛІДЖЕННЯХ. *THEORETICAL AND PRACTICAL ASPECTS OF MODERN SCIENTIFIC RESEARCH*. 2021. URL: <https://doi.org/10.36074/logos-30.04.2021.v1.11> (дата звернення: 23.11.2024).
 10. Система управління базами даних: сучасні тенденції у розробці. *FoxmindEd*. URL: <https://foxminded.ua/systema-upravlinnia-bazamy-danykh/> (дата звернення: 23.11.2024).
 11. Bradbury J. D., Guadagno R. E. Documentary narrative visualization: Features and modes of documentary film in narrative visualization. *Information Visualization*. 2020. Т. 19, № 4. С. 339–352. URL: <https://doi.org/10.1177/1473871620925071> (дата звернення: 23.11.2024).
 12. Doroshenko A. Y., Tulika E. M. Parallel algorithms optimization using Actor Model. *PROBLEMS IN PROGRAMMING*. 2020. № 2-3. С. 126–137. URL: <https://doi.org/10.15407/pp2020.02-03.126> (дата звернення: 23.11.2024).
 13. Review of multi-criteria decision-making for sustainable decentralized hybrid energy systems / S. Das та ін. *Renewable and Sustainable Energy Reviews*. 2024. Т. 202. С. 114676. URL: <https://doi.org/10.1016/j.rser.2024.114676> (дата звернення: 23.11.2024).
 14. bigData: hadoop. JavaRush. URL: <https://javarush.com/ua/quests/lectures/ua.questhibernate.level20.lecture03> (дата звернення: 03.06.2024).
 15. overview - spark 3.5.1 documentation. Apache Spark™ - Unified Engine for large-scale data analytics. URL: <https://spark.apache.org/docs/latest/> (дата звернення: 03.06.2024).

16. what Is NoSQL? NoSQL Databases Explained. MongoDB. URL: <https://www.mongodb.com/resources/basics/databases/nosql-explained> (дата звернення: 03.06.2024).
17. Deep Learning Based Recommender System / S. Zhang та ін. *ACM Computing Surveys*. 2019. Т. 52, № 1. С. 1–38. URL: <https://doi.org/10.1145/3285029> (дата звернення: 23.11.2024).
18. what is the best way to evaluate an algorithm's performance for Operations Research?. LinkedIn: Log In or Sign Up. URL: <https://www.linkedin.com/advice/0/what-best-way-evaluate-algorithms-performance> (дата звернення: 03.06.2024).
19. Hosseinzadeh N. Power System Strength: Evaluation Methods, Best Practice, Case Studies, and Applications. Institution of Engineering & Technology, 2023.
20. Firestone W. A., Donaldson M. L. Teacher evaluation as data use: what recent research suggests. *Educational Assessment, Evaluation and Accountability*. 2019. Т. 31, № 3. С. 289–314. URL: <https://doi.org/10.1007/s11092-019-09300-z> (дата звернення: 23.11.2024).
21. Efficient and accurate compound scaling for convolutional neural networks / C. Lin та ін. *Neural Networks*. 2023. URL: <https://doi.org/10.1016/j.neunet.2023.08.053> (дата звернення: 23.11.2024).
22. Sandhu D. Student Resource Pack: My Learning Place. Independently Published, 2019.
23. Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge University Press, 2018. 491 с.
24. Best Practices for Web Development using Grails and Django / L. P. Mora Murguía та ін. *Research in Computing Science*. 2014. Т. 78, № 1. С. 9–20. URL: <https://doi.org/10.13053/rscs-78-1-1> (дата звернення: 23.11.2024).
25. A normal cloud model-based decision making method under multi-granular probabilistic linguistic environment for evaluating of wetland ecosystem

- services / L. Weng та ін. *Expert Systems with Applications*. 2023. С. 120202.
URL: <https://doi.org/10.1016/j.eswa.2023.120202> (дата звернення: 23.11.2024).
26. Lemenkova P. PROCESSING OCEANOGRAPHIC DATA BY PYTHON LIBRARIES NUMPY, SCIPY AND PANDAS. *Aquatic Research*. 2019. С. 73–91. URL: <https://doi.org/10.3153/ar19009> (дата звернення: 24.11.2024).
 27. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, Incorporated, 2018.
 28. Srivastava A. K. Django , The Python Web Framework. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. 2022. Т. 06, № 05. URL: <https://doi.org/10.55041/ijssrem13183> (дата звернення: 24.11.2024).
 29. Building RESTful Python Web Services. Packt Publishing - ebooks Account, 2016. 335 с.
 30. Ororbia A. G., Mali A. Backprop-Free Reinforcement Learning with Active Neural Generative Coding. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022. Т. 36, № 1. С. 29–37. URL: <https://doi.org/10.1609/aaai.v36i1.19876> (дата звернення: 24.11.2024).

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

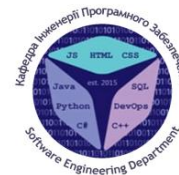


ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ

ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота

**«МЕТОДИКА ПІДБОРУ АВТО З ВИКОРИСТАННЯМ РЕКОМЕНДАЦІЙНИХ
АЛГОРИТМІВ»**

Виконав: Студент групи ПДМ-61, Лебідь Максим Максимович

Керівник: к.т.н., доц., доцент кафедри ІТ, Трінтіна Наталя Альбертівна

Київ - 2025



МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи – оптимізація процесу вибору автомобіля шляхом розробки методики, що зменшує часові показники за рахунок використання рекомендаційних систем.

Об'єкт дослідження – процес підбору автомобілів.

Предмет дослідження – методика підбору автомобілів з використанням рекомендаційних алгоритмів.



АНАЛІЗ СПОСОБІВ ПІДБОРУ АВТО

Метод	Об'єктивність	Рівень експертності	Необхідний час	Доступність 24/7	Емоційна складова
Особисті рекомендації	Суб'єктивна	Низький(непрофесійний)	Швидкий (хвилини-години)	Ні	Висока (вплив близьких людей)
Тест-драйв	Суб'єктивна (вплив продавця)	Середній (знання продавця)	Помірний (години)	Ні (робочі години)	Можливий вплив емоцій клієнта
Друковані видання та каталоги	Об'єктивна	Високий (експертні дані)	Варіативний (години-дні)	Обмежена	Немає
Експертні рекомендації	Залежить від експерта	Високий (професійний досвід)	Помірний (години)	Ні (за домовленістю)	Можливо врахування емоцій
Рекомендаційні системи (загальні)	Висока (аналіз даних)	Алгоритмічна	Швидкий (хвилини)	Так	Немає
Рекомендаційні системи з використанням ІІІ	Дуже висока (глибокий аналіз даних)	Висока (алгоритми ІІІ)	Миттєвий (реальний час)	Так	Немає ³

АНАЛІЗ СУЧАСНИХ ОНЛАЙН-ПЛАТФОРМ ТА РОЗРОБЛЕНОЇ МЕТОДИКИ ДЛЯ ПІДБОРУ АВТОМОБІЛІВ

Платформа / Методика	Базовий пошук та фільтрація	Рекомендаційні алгоритми	Використання ІІІ	Швидкий та точний підбір за індивідуальними параметрами
AutoScout24	Так	Частково	Ні	Частково забезпечується, але не повністю
AUTO.RIA.com	Так	Частково	Ні	Частково забезпечується, але обмежено
OLX.ua	Так	Ні	Ні	Не забезпечується повністю
RST.ua	Так	Ні	Ні	Не забезпечується повністю
Розроблена методика	Так	Так	Так	Так, повністю забезпечується

СТРУКТУРА РЕКОМЕНДАЦІЙНОГО АЛГОРИТМУ



5

ІНТЕГРАЦІЯ ТА ФУНКЦІОНУВАННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ



6

МАТЕМАТИЧНА МОДЕЛЬ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ІІІ

1. Розрахунок прогнозованого рейтингу автомобіля:

$$Z = \sum_i \beta_i \times K_{i_{\text{авт}}}$$

Де:

- β_i — вагові коефіцієнти.
- $K_{i_{\text{авт}}}$ — нормалізовані значення критеріїв.

2. Час обробки:

$$T_{\text{ІІІ}} = T_{\text{навч}} + T_{\text{прогн}}$$

Де:

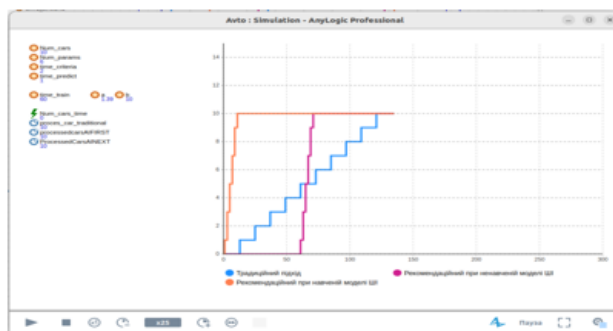
- $T_{\text{навч}} = a \cdot n^2 + b$

Де:

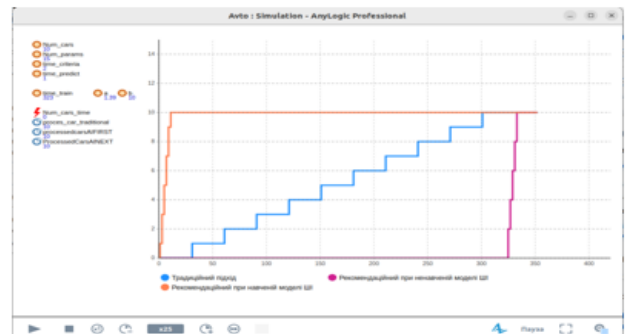
- n : кількість пріоритетних критеріїв;
- a : коефіцієнт, що визначає вплив взаємодій між критеріями на час;
- b : постійний базовий час, незалежний від кількості критеріїв.
- $T_{\text{прогн}}$ — час прогнозування для одного автомобіля.

7

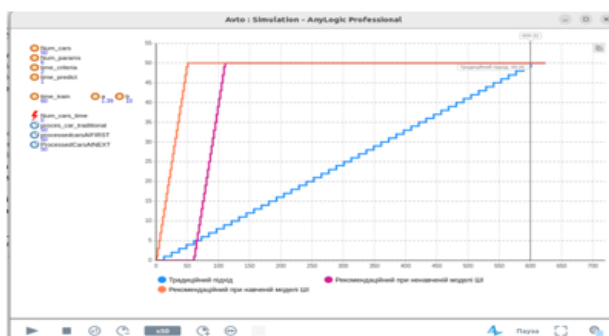
ГРАФІЧНИЙ АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ СЦЕНАРІЇВ ANYLOGIC ДЛЯ РІЗНИХ ПАРАМЕТРІВ



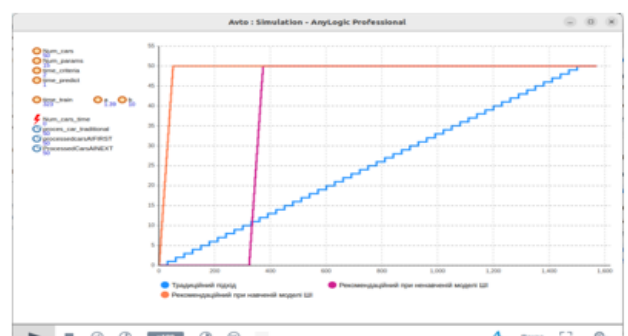
Симуляція 1: Мало Авто, Мало Параметрів



Симуляція 2: Мало Авто, Багато Параметрів



Симуляція 3: Багато Авто, Мало Параметрів



Симуляція 4: Багато Авто, Багато Параметрів

8

ПОРІВНЯЛЬНА ТАБЛИЦЯ РЕЗУЛЬТАТІВ ЕКСПЕРИМЕНТІВ В ANYLOGIC

Симуляція	Традиційний Підхід	Не навчений ІІІ	Навчений ІІІ	Коеф. Традиц. Не навч.	Коеф. Традиц. Навч.
Cars 10, Params 6	120	70	10	1.71	12.0
Cars 10, Params 15	300	329	10	0.91	30.0
Cars 50, Params 6	600	110	5	5.45	120.0
Cars 50, Params 15	1500	373	5	4.02	300.0

9

ІНТЕРФЕЙС РЕКОМЕНДАЦІЙНОГО АЛГОРИТМУ В DJANGO

Назва	Бренд	Ціна	Пробіг	Вік	Витрати пального	Надійність	Комфорт	Рейтинг
Nissan 2008	Nissan	25234.0	161909.0	16.0	9.0	2.0	2.0	12.15
Toyota 2009	Toyota	32905.0	106933.0	15.0	7.0	5.0	7.0	13.12
Nissan 2010	Nissan	46534.0	68920.0	14.0	12.0	8.0	10.0	14.35
BMW 2016	BMW	27619.0	28620.0	8.0	10.0	6.0	7.0	13.55
BMW 2017	BMW	21878.0	66843.0	7.0	10.0	8.0	4.0	12.95
Chevrolet 2015	Chevrolet	44938.0	77545.0	9.0	6.0	2.0	6.0	12.65
Audi 2006	Audi	12505.0	102397.0	18.0	7.0	6.0	3.0	12.38
Honda 2021	Honda	9641.0	27919.0	3.0	6.0	2.0	7.0	12.82
Honda 2016	Honda	25349.0	130714.0	8.0	5.0	3.0	9.0	13.19
Mercedes 2006	Mercedes	18121.0	73314.0	18.0	12.0	3.0	6.0	13.39
Toyota 2008	Toyota	35807.0	59473.0	16.0	6.0	6.0	5.0	12.69
Chevrolet 2017	Chevrolet	32184.0	42423.0	7.0	9.0	2.0	8.0	13.32
Hyundai 2018	Hyundai	6028.0	42205.0	6.0	11.0	7.0	4.0	12.99

Доступні авто

Ціна

Пробіг

Вік

12249

172224

19

Витрати пального

Надійність

Комфорт

6

7

10

Знайти авто

Топ-5 рекомендованих авто

Назва	Бренд	Ціна	Пробіг	Вік	Витрати пального	Надійність	Комфорт	Рейтинг
Kia 2005	Kia	37269.0	172224.0	19.0	6.0	7.0	10.0	13.73
Toyota 2021	Toyota	94021.0	73389.0	3.0	10.0	10.0	5.0	13.73
Chevrolet 2013	Chevrolet	21816.0	186188.0	11.0	9.0	6.0	9.0	13.74
Toyota 2013	Toyota	17173.0	187386.0	11.0	14.0	1.0	8.0	13.73
Ford 2015	Ford	43683.0	102967.0	9.0	13.0	6.0	7.0	13.73

Прогнозований рейтинг вашого авто

Прогнозований рейтинг для ваших критеріїв: 13.73

Результат пошуку

Ціна

Пробіг

Вік

7000

70000

9

Витрати пального

Надійність

Комфорт

10

10

8

Розрахувати рейтинг

Результат

Статус: success

Сирій рейтинг: -740.2787947102606

Нормалізований рейтинг: 13.856199661842098

Розрахунок рейтингу для API

10

ВИСНОВКИ

1. Проведено аналіз традиційних та сучасних методів підбору автомобілів. Виявлено, що традиційні методи, такі як особисті рекомендації та відвідування автосалонів, мають обмеження щодо інформації та витрат часу. Сучасні підходи з використанням рекомендаційних алгоритмів дозволяють персоналізувати пропозиції та спростити процес вибору для користувача.
2. Досліджено сучасні онлайн-платформи для підбору автомобілів, такі як AutoScout24, AUTO.RIA.com, OLX.ua та інші. Встановлено, що вони надають базовий функціонал фільтрації та пошуку, але не повністю задовольняють потреби у швидкому та точному підборі за індивідуальними параметрами.
3. Розроблено методику, яка включає збір і обробку великих обсягів даних про автомобілі, використання сучасних інструментів для роботи з ними, аналіз та вибір оптимальних алгоритмів машинного навчання. Адаптовано обрані алгоритми для задачі підбору з урахуванням специфіки предметної області.
4. Розроблено програмну систему рекомендацій мовою Python з використанням фреймворку Django. Реалізовано модулі обробки та аналізу даних, інтегровано модель машинного навчання, інтерфейс користувача та API. Забезпечено безпеку, валідацію даних та ефективну взаємодію з базою даних PostgreSQL.
5. Проведено серію експериментів з моделюванням у середовищі AnyLogic. Результати показали, що використання навченої моделі штучного інтелекту скорочує час вибору автомобіля у 59 разів порівняно з звичайними рекомендаційними методами, підтверджуючи високу ефективність методики.

11

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Статті:

1. Лебідь М. М. Оптимізація методики підбору автомобілів за допомогою рекомендаційних алгоритмів // Зв'язок. – подана до друку.

Тези доповідей:

1. Лебідь М. М., Трінтіна Н. А. Використання інтелектуального аналізу великих обсягів даних для оптимізації стратегій маркетингу та підвищення ефективності обробки даних // Застосування програмного забезпечення в ІКТ : Всеукр. науково-техн. конф., Київ, 24 квіт. 2024 р. – Київ, 2024. – С. 147.
2. Лебідь М. М., Трінтіна Н. А. Застосування рекомендаційних алгоритмів для автоматизації процесу підбору автомобілів // Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу : II всеукр. науково-техн. конф., Київ, 18 листоп. 2024 р. – Київ, 2024. – подана до друку.

ДОДАТОК Б.

ЛІСТИНГИ ОСНОВНИХ МОДУЛІВ

<pre>settings.py from pathlib import Path import os # Build paths inside the project like this: BASE_DIR / 'subdir'. BASE_DIR = Path(__file__).resolve().parent.parent # Quick-start development settings - # unsuitable for production SECRET_KEY = os.getenv('DJANGO_SECRET_KEY', 'django-insecure- 8z(5eo#9+cg9&u%mz!s62v_ecp4eak- py=-k^eud1-xrx+9+io') DEBUG = True ALLOWED_HOSTS = ['127.0.0.1', 'localhost'] # Application definition INSTALLED_APPS = ['django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles', 'recommendation', 'common_templates',] MIDDLEWARE = [</pre>	<pre>'django.middleware.security.Security Middleware', 'django.contrib.sessions.middleware.S essionMiddleware', 'django.middleware.common.Common Middleware', 'django.middleware.csrf.CsrfViewMid dleware', 'django.contrib.auth.middleware.Auth enticationMiddleware', 'django.contrib.messages.middleware. MessageMiddleware', 'django.middleware.clickjacking.XFra meOptionsMiddleware',] ROOT_URLCONF = 'car_recommendation.urls' TEMPLATES = [{ 'BACKEND': 'django.template.backends.django.Dja ngoTemplates', 'DIRS': [], 'APP_DIRS': True, 'OPTIONS': { 'context_processors': ['django.template.context_processors.d ebug', 'django.template.context_processors.re quest',</pre>
--	---

```

        'django.contrib.auth.context_processors.auth',

        'django.contrib.messages.context_processors.messages',
    ],
},
]

WSGI_APPLICATION = 'car_recommendation.wsgi.application'

# Database
DATABASES = {
    'default': {
        'ENGINE':
            'django.db.backends.postgresql_psycopg2',
        'NAME': 'local_multicora',
        'USER': 'main_user',
        'PASSWORD': 'Qwerty321',
        'HOST': '192.168.0.124',
        'PORT': '',
    }
}

# Password validation
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
            'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
            'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
]

# Internationalization
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Europe/Kyiv'

USE_I18N = True

USE_TZ = True

# Static files
STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / "static"]
STATIC_ROOT = BASE_DIR / "staticfiles"

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

Urls.py
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path("", include('recommendation.urls')),
]

models.py

```

```
from django.db import models
```

```
class Car(models.Model):
```

```
    name = models.CharField(max_length=100)
    brand = models.CharField(max_length=100)
    price = models.FloatField() # Ціна
    mileage = models.FloatField() # Пробіг
    age = models.FloatField() # Вік
    fuel_efficiency = models.FloatField() #
        Витрати пального
    reliability = models.FloatField() #
        Надійність
    comfort = models.FloatField() #
        Комфорт
    rating = models.FloatField(null=True,
        blank=True) # Розрахований
        рейтинг
```

```
    def __str__(self):
    return self.name
```

```
urls.py
```

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path("", views.home, name='home'),
    path('recommend/', views.recommend,
        name='recommend'),
    path('cars/', views.car_list,
        name='car_list'),
    path('recommend_car/',
        views.recommend_car,
        name='recommend_car'),
    path('api/calculate_rating/',
        views.api_calculate_rating,
        name='api_calculate_rating'), # API
        виклик
    path('api_page/', views.api_page,
        name='api_page'), # Сторінка API
```

```
]
```

```
Views.py
```

```
import pickle
import numpy as np
from django.shortcuts import render
from .models import Car
from django.db.models import F
from django.db.models.functions import Abs
from django.views.decorators.csrf import csrf_exempt
from django.http import JsonResponse
import json
```

```
# Завантаження моделі
```

```
def load_model():
    with open('model.pkl', 'rb') as file:
        data = pickle.load(file)
    return data['model'], data['scaler']
```

```
model, scaler = load_model()
```

```
def normalize_features(features):
```

```
    min_values = np.array([5000, 10000, 1,
        5, 1, 1]) # Мінімальні можливі
        значення для критеріїв
```

```
    max_values = np.array([50000, 200000,
        20, 15, 10, 10]) # Максимальні
        можливі значення для критеріїв
```

```
    normalized_features = (features -
        min_values) / (max_values -
        min_values)
```

```
    return normalized_features
```

```
# Обчислення нормалізованого
    рейтингу
```

```
def calculate_normalized_rating(features):
```

```

normalized_features =
    normalize_features(features)

scaled_features =
    scaler.transform(normalized_features)

raw_rating =
    model.predict(scaled_features)[0]

# Нормалізація рейтингу до [1, 100]
min_rating = -1000 # Мінімальний
    сирий рейтинг, зафіксований під
    час навчання моделі

max_rating = 1000 # Максимальний
    сирий рейтинг

normalized_rating = 1 + (raw_rating -
    min_rating) / (max_rating -
    min_rating) * 99 # Нормалізація до
    [1, 100]

return normalized_rating, raw_rating

# Головна сторінка
def home(request):
    return render(request, 'home.html')

# Рекомендація рейтингу для введених
    параметрів
def recommend(request):
    if request.method == 'GET':
        try:
            # Отримання параметрів із
                запиту
            price =
                float(request.GET.get('price', 0))
            mileage =
                float(request.GET.get('mileage', 0))
            age = float(request.GET.get('age',
                0))
            fuel_efficiency =
                float(request.GET.get('fuel_efficiency'
                , 0))
            reliability =
                float(request.GET.get('reliability', 0))
            comfort =
                float(request.GET.get('comfort', 0))

            # Підготовка даних
            features = np.array([price,

```

```

mileage, age, fuel_efficiency,
    reliability, comfort]])

        normalized_rating, raw_rating =
            calculate_normalized_rating(features)

        return render(request,
            'recommend_result.html', {

                'raw_rating': raw_rating,

                'normalized_rating':
                    normalized_rating

            })

        except Exception as e:

            return JsonResponse({'error':
                'Invalid input data. Please check your
                inputs.'}, status=400)

# Список всіх автомобілів
def car_list(request):
    cars = Car.objects.all() # Отримуємо
        всі записи з таблиці Car

    return render(request, 'car_list.html',
        {'cars': cars})

# Рекомендація автомобілів
def recommend_car(request):
    recommended_cars = []
    predicted_rating = None
    input_values = {
        'price': request.GET.get('price', ''),
        'mileage': request.GET.get('mileage',
            ''),
        'age': request.GET.get('age', ''),
        'fuel_efficiency':
            request.GET.get('fuel_efficiency', ''),
        'reliability':
            request.GET.get('reliability', ''),
        'comfort': request.GET.get('comfort',
            ''),
    }

    if request.method == 'GET' and
        all(input_values.values()):
        try:
            # Отримуємо введені значення

```

```

        price = float(input_values['price'])
        mileage = float(input_values['mileage'])
        age = float(input_values['age'])
        fuel_efficiency = float(input_values['fuel_efficiency'])
        reliability = float(input_values['reliability'])
        comfort = float(input_values['comfort'])

        # Формуємо вектор введенних значень
        features = np.array([[price, mileage, age, fuel_efficiency, reliability, comfort]])

        predicted_rating, _ = calculate_normalized_rating(features)
        # Обчислюємо рейтинг

        # Знаходимо топ-5 авто за найближчим рейтингом
        recommended_cars = Car.objects.annotate(
            rating_difference=Abs(F('rating') - predicted_rating) # Абсолютна різниця між рейтингами
        ).filter(
            rating__isnull=False # Виключаємо авто без рейтингу
        ).order_by('rating_difference')[:5]
        # Сортуюмо за різницею і беремо топ-5

        except Exception as e:
            print(f"Error: {e}")
            recommended_cars = []

    return render(request, 'recommend_car.html', {
        'cars': recommended_cars,
        'input_values': input_values,
        'predicted_rating': predicted_rating,
    })

# Оновлення рейтингів у базі даних
def calculate_ratings(request):

cars = Car.objects.all()
raw_ratings = []
for car in cars:
    features = np.array([car.price, car.mileage, car.age, car.fuel_efficiency, car.reliability, car.comfort])
    normalized_rating, raw_rating = calculate_normalized_rating(features)
    raw_ratings.append((car, normalized_rating, raw_rating))

# Збереження нормалізованих рейтингів у базу
for car, normalized_rating, _ in raw_ratings:
    car.rating = normalized_rating
    car.save()

return JsonResponse({'status': 'Ratings updated and normalized successfully!'})

def api_page(request):
    return render(request, 'api_calculate_rating.html')

from django.views.decorators.csrf import csrf_exempt
from django.http import JsonResponse
from django.shortcuts import render
import json

@csrf_exempt
def api_calculate_rating(request):
    if request.method == 'POST':
        try:
            # Отримання даних із POST-запиту
            data = json.loads(request.body)
            price = float(data.get('price', 0))
            mileage = float(data.get('mileage', 0))
            age = float(data.get('age', 0))
            fuel_efficiency =

```

```

float(data.get('fuel_efficiency', 0))
    reliability =
float(data.get('reliability', 0))
    comfort = float(data.get('comfort',
0))
    # Підготовка вхідних даних
    features = np.array([[price,
mileage, age, fuel_efficiency,
reliability, comfort]])
    normalized_rating, raw_rating =
calculate_normalized_rating(features)
    # Повертаємо JSON-відповідь
    return JsonResponse({
        'status': 'success',
        'raw_rating': raw_rating,
        'normalized_rating':
normalized_rating
    })
except Exception as e:
    return JsonResponse({'status':
'error', 'message': str(e)}, status=400)
elif request.method == 'GET':
    # Повертаємо сторінку для
взаємодії з API
    return render(request,
'api_calculate_rating.html')
else:
    return JsonResponse({'status': 'error',
'message': 'Method not allowed.'}, status=405)

```