Київський національний університет імені Тараса Шевченка

Економічний факультет

Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Моделювання цін на вторинному автомобільному ринку за допомогою алгоритмів машинного навчання

студентки 4 курсу

спеціальності 051 «Економіка»

ОеП «Економічна кібернетика»

денної форми навчання

Сілюк Дарини Миколаївни

Науковий керівник:

кандидат економічних наук, доцент

Федоренко Ірина Костянтинівна

Засвідчую, що в цій роботі немає запозичень із праць інших авторів без відповідних посилань

Студентка _____

Роботу допущено до захисту перед ЕК рішенням кафедри економічної кібернетики від 12 червня 2025 р., протокол № 15

Завідувач кафедри: доктор економічних наук, професор Ляшенко Олена Ігорівна _____

КИЇВ – 2025

РЕФЕРАТ

Кваліфікаційна робота бакалавра містить: 63 ст., 16 рис., 2 табл., 30 джерел.

Ключові слова: автомобільний ринок, Індекс прозорості ринку, моделювання, машинне навчання, розвідувальний аналіз, Python, прогнозування, вживані автомобілі.

Об'єкт дослідження: процес ціноутворення на ринку вживаних автомобілів.

Мета дослідження: розробка моделі прогнозування ціни вживаних автомобілів на основі реальних характеристик.

Методи дослідження:

- Аналіз статистичних даних.
- Методи машинного навчання (регресійні моделі, розвідувальний аналіз даних).
- Методи візуалізації та обробки даних (бібліотеки Python: pandas, matplotlib, seaborn, sklearn)

Наукова новизна, теоретична значимість дослідження: у ході дослідження було запропоновано підхід до моделювання ринкової вартості вживаних автомобільних транспортів, використовуючи реальні дані українського ринку. Машинне навчання дозволило об'єктивно визначити ключові фактори ціноутворення, а також продемонструвати потенціал використання цифрових інструментів в економічному аналізі та прогнозуванні.

Практична цінність: модель може бути використана онлайн-платформами для оцінки вартості транспортних засобів, страховими компаніями та банками для перевірки реальної ціни авто, покупцями та продавцями для прийняття обґрунтованих рішень, а також для виявлення шахрайських дій на ринку.

RESUME

Taras Shevchenko National University of Kyiv,

Faculty of Economics, Department of Economic Cybernetics

Key words: automotive market, market transparency index, modeling, machine learning, exploratory analysis, Python, forecasting, used cars.

The graduation research of student Daryna Siliuk **deals with** the development and application of a Random Forest Regression model to predict used car prices based on their technical characteristics and market data.

The work is interesting for researchers, data analysts, and professionals in the automotive and used car markets who seek accurate price prediction models to improve decision-making and market analysis.

Pages 63, tables 2, bibliog. 30.

ЗМІСТ

ВСТУП

Тема дослідження є актуальною, тому що автомобільний ринок є одним з ключових сегментів економіки і чинить суттєвий вплив на споживчу поведінку. Особливо активним залишається саме вторинний ринок, так як споживачі зацікавлені в тому, щоб отримати якісний автомобіль за доступною ціною, а продавці намагаються встановити конкуренту ціну.

На формування вартості автомобіля впливають багато факторів: рік випуску, пробіг, технічний стан, тип палива, модель, комплектація, регіон продажу тощо. В Україні значна частка автомобільного ринку перемістилася в онлайн-простір, у зв’язку з війною, тому актуальність точної оцінки вартості транспорту зросла ще більше.

Моделі ціноутворення вже досліджувалися такими науковцями як Parth Bhatnagar, Gururaj Harinahalli Lokesh, J Shreyas, Francesco Flammini, Shivansh Gautam [1], Suhas Maddali [2], Sanchith Venkatesan, Shivam Handa, Baskar M [3] та Sameerchand Pudaruth [4], проте їхні дослідження базуються на даних західних країн і не враховують особливості українського ринку. Тому необхідно було створити модель, адаптовану під ринок Україні.

Об’єктом дослідження є процес ціноутворення на ринку вживаних автомобілів.

Предметом дослідження є прогнозування ринкової вартості вживаних автомобілів на основі технічних характеристик.

Основні завдання дослідження:

1. Аналіз підходів до оцінки вартості вживаних авто.
2. Збір та обробка даних.
3. Розвідувальний аналіз даних.
4. Побудова моделі для прогнозування за допомогою машинного навчання.
5. Оцінка точності прогнозу та основних факторів впливу на ціноутворення.

Методи дослідження:

- Описова статистика.
- Методи машинного навчання (регресійні моделі, дерева рішень, RandomForest).
- Методи візуалізації та обробки даних (бібліотеки Python: pandas, matplotlib, seaborn, sklearn).

Наукова новизна полягає в адаптації сучасних моделей машинного навчання до специфіки українського ринку вживаних автомобілів.

Практичною цінністю є можливість впровадити модель в онлайн-платформи, щоб допомагати користувачам визначати реальну ціну автомобіля враховуючи всі фактори.

Результати впровадженої роботи презентувалися на XXIII Міжнародній науково-практичній конференції “Шевченківська весна 2025. Економіка України 2025: нові вектори розвитку в умовах глобальних трансформацій” [5].

Основною інформаційною базою дослідження була платформа auto.ria [6], звідки було отримано дані для подальшої роботи з ними.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків і списку використаних джерел. **Перший розділ** описує загальні тенденції та проблематику на ринку вживаних автомобілів. **Другий розділ** присвячено методології прогнозної аналітики та огляду сучасних моделей машинного навчання. **Третій розділ** описує процес побудови, навчання та тестування моделі прогнозування цін на основі заданих характеристик.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА РИНКУ ВЖИВАНИХ АВТОМОБІЛІВ

1.1. Проблематика автомобільного ринку

Автомобільний ринок відіграє важливу роль у глобальній економіці. У 2024 році було виготовлено 92,5 млн одиниць автотранспорту, з них 67,7 млн — легкові автомобілі. Найбільшими виробниками на той момент були Китай, США та Японія.

З виходом нових моделей, власники транспорту більш схильні продавати старі авто та купувати нові. Таким чином, виникає проблема пошуку оптимальної ціни, для вживаного автомобіля. Точне прогнозування та моделювання ціни можна зробити за допомогою алгоритмів машинного навчання. Це включає визначення ключових факторів, що впливають на ціноутворення, а також навчання алгоритму, який буде передбачати оптимальну ціну [7].

Така модель може вирішити ряд проблем:

1. Забезпечити прозорість цін на вторинному ринку.
2. Допомогти покупцям і продавцям приймати обґрунтовані рішення.
3. Зменшити інформаційну асиметрію між продавцями та покупцями.
4. Оптимізувати бізнес-процеси для дилерів та онлайн-платформ продажу авто.
5. Розробити автоматизований інструмент, який стане в нагоді для страхових компаній та банків.
6. Виявити аномальні цінові пропозиції та потенційне шахрайство.

1.2. Український ринок вживаних автомобілів

Після введення військового стану став актуальним закон “Про мобілізаційну підготовку та мобілізацію”. У статті 6 “Військово-транспортний обов’язок” визначено основні засади та порядок виконання обов’язку надання власного транспортного засобу для забезпечення потреб ЗСУ. Це спричинило активне переоформлення автомобілів, а також їх продажі [8].

У кінці 2024 року Інститут досліджень авторинку спостерігав за активністю на різних платформах для продажу. Таким чином було виявлено, що все менше продавців продовжують використовувати “класичні” сайти оголошень. У період активного використання соціальних мереж, таких як TikTok, Instagram, Facebook, Telegram, користувачі мають змогу безкоштовно розміщувати контент, збільшувати аудиторію і таким чином легше продавати авто [9].

Держава також сприяє інтернет-продажам. У кінці 2023 року Дія запустила послугу перереєстрації авто онлайн. Тепер сторонам купівлі-продажу не обов’язково йти у сервісний центр МВС та платити великі комісії за обслуговування. Достатньо лише заповнити договір і заяву у застосунку, та почекати годину. Через деякий час новому власнику надійде пакет усіх документів, які будуть офіційно затверджені МВС [10].

1.3. Вплив економічних факторів на ціноутворення на прикладі США

Для людей, які перепродають автівки, критично важливо не лише звертати увагу на ціну купівлі, а й враховувати ціну майбутнього продажу. Власники можуть вплинути на стан транспорту, пробіг, якість обслуговування, але не на економічну ситуацію в країні та в світі. Попри це, останній фактор є не менш важливим, так як стан економіки чинить сильний вплив на вартість.

У 2025 році компанія Manheim провела дослідження, у якому виявила, що зниження цін на ринку вживаних авто у США значною мірою було спричинено економічними чинниками. Індекс вартості вживаних автомобілів Manheim у березні 2025 року знизився на 1,1 % порівняно з тим же місяцем минулого року. Причиною цього стало послаблення економічної активності, а також змінилася

структура пропозиції: збільшилася частка старих автомобілів через зменшення кількості лізингових контрактів.

Падіння Індексу споживчих настроїв розрахованого Університетом Мічигану у березні 2025 року на 10,5 % свідчить про значне погіршення оцінок щодо економічних умов та майбутніх очікувань. Чим менше користувачі впевнені в стабільності, тим нижчим буде рівень їх готовність витрачати кошти на великі покупки. Таким чином, оцінки щодо умов купівлі транспортних засобів знизилися до найнижчого рівня порівняно з останніми 2 роками.

Важливим фактором є інфляційна невизначеність. У 2025 році американці очікували значне підвищення інфляції (на 4,9 %), що свідчить про недовіру та невпевненість у стабільності. Це також призводить до зменшення попиту і впливає на ціноутворення. Зниження купівельної спроможності населення через інфляцію спричиняє переорієнтацію споживачів з нових автомобілів на вживані, як більш доступну альтернативу. Тоді продавці підвищують ціни, щоб захистити кошти від знецінення, а покупці активніше інвестують у автомобілі як у матеріальний актив, щоб зберегти гроші під час інфляції.

Крім того, у 2025 році на ціноутворення вживаних автомобілів у США впливали мита, структура ринку, а також обмежена пропозиція. Усі чинники комплексно призвели до зниження цін на автомобілі, попри очікування щодо сезонного підвищення вартості [11].

1.4. Вплив економіки на споживчі настрої в Україні

У квітні 2025 року Центр економічної стратегії оновив статтю «Трекер економіки України під час війни». Статті почали публікуватися з початку повномасштабного вторгнення і з того часу регулярно оновлюються.

Українська економіка поступово відновлюється після повномасштабного вторгнення і вже у 2024 році реальний ВВП становив 77 % від показника у 2021 році.

Вплив на автомобільний ринок: через повільні темпи відновлення економічної активності споживчі витрати все ще лишаються обмеженими і користувачі рідше роблять великі покупки [12].

Через війну багато виробництв було зруйновано і ворог продовжує завдавати ударів по важливій інфраструктурі. Якщо у 2023 році вдалося опанувати прискорення інфляції, то вже у 2024 році вона знову почала зростати через несприятливі економічні та погодні умови. Це спричинило зростання цін на продукти харчування у лютому 2024 року. У березні 2025 року НБУ підвищив облікову ставку до 15,5%, щоб стримати інфляцію.

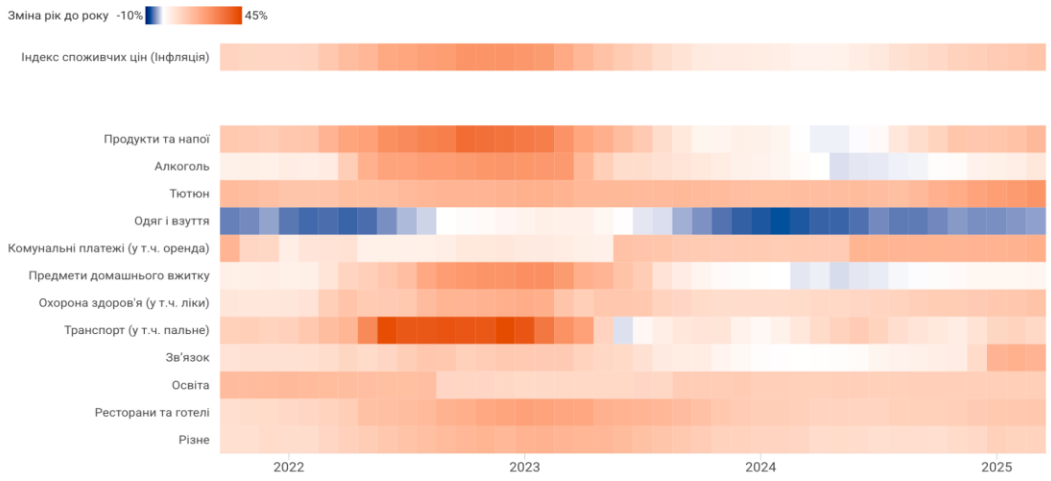

Рис. 1.1. Теплокарта інфляції.

Джерело: [12]

Вплив на автомобільний ринок: на теплокарті (рис. 1.1.) видно, що ціни на транспорт значно зросли з початку війни та продовжували зростати ще протягом 1 кварталу 2023 року. Потім темп зростання знизився та стабілізувався. Висока інфляція, зокрема на продукти, означає, що витрати на великі покупки, як-от автомобілі, відкладаються. Підвищення облікової ставки НБУ робить кредити дорожчими, що обмежує доступність автокредитування [12].

У 2025 році збільшилися податкові надходження. Внутрішній ПДВ зріс на 17 % у лютому того ж року, а ПДВ на імпорт зріс на 7 %. Акцизні податки зросли на 70 % порівняно з минулим роком.

Реальні загальні видатки (скориговані на інфляції) у 2024 році залишилися на тому ж рівні, як і минулого року. 65 % загальних видатків спрямовується на військові потреби.

Вплив на автомобільний ринок: зростання податкових надходжень, зокрема ПДВ, може свідчити про активізацію внутрішнього ринку. Це створює позитивний сигнал для автомобільного сектору, але зростання акцизів, особливо на пальне, може підвищити вартість експлуатації автомобілів, що зменшить інтерес до купівлі [12].

На початку повномасштабного вторгнення не було ні попиту, ні пропозиції на робочу силу. У 2025 році ситуація покращилася, але все ще зберігається нижча активність на ринку праці, порівняно з довоєнними роками. За результатами дослідження агенції Info Sapiens рівень безробіття в Україні зменшився до 12 % у березні 2025 року. Попри це, збільшився показник рівня бідності та досяг майже 25 %.

Вплив на автомобільний ринок: низька активність на ринку праці та підвищення показника бідності негативно впливає на попит у сфері транспорту — споживачі не можуть дозволити покупку дорогих речей [12].

Усі власні надходження починаючи з повномасштабного вторгнення йдуть на потреби війни, цивільні видатки фінансуються за рахунок зовнішнього фінансування. У 2024 році іноземна допомога покрила лише 73 % необхідних витрат і частина дефіциту покривалася за рахунок облігацій внутрішньої державної позики.

Вплив на автомобільний ринок: іноземна допомога дозволяє фінансувати соціальні програми та зарплати, зберігаючи споживчу активність. Надходження коштів допомагають стримувати економічну дестабілізацію, що опосередковано підтримує попит на автомобілі, особливо бюджетного сегменту [12].

1.5. Ціноутворення на українському ринку

Цінову картину досліджував Інститут досліджень авторинку у лютому 2025 року. Визначено, що середня ціна вживаних автомобілей у січні 2025 року була

6 000 $, що вище за попередні значення. Областями з найближчими цінами до середньої були Рівненська (6 300 $), Одеська (6 200 $), Тернопільська (5 800 $) та Вінницька (5 700 $).

Територіями з найвищими середніми цінами є Київ та Київська область (8 600 $), що цілком зрозуміло, адже там високий попит на автомобілі, високий рівень доходу та вища активність автодилерів, ніж у інших регіонах. Львівська область має дещо нижчу середню ціну — 8 000 $. У цьому регіоні також спостерігається високий попит, особливо на транспортні засоби з Європи. Закарпатська область — 6 800 $. Схожа ситуація як і у Львівській області: високий попит на автомобілі з інших країн через близьке розташування до кордону.

Найнижчі середні ціни спостерігаються у Херсонській області (2 650 $), що спричинено часткою окупації та близьким розташуванням до зони бойових дій. У Донецькій та Луганській областях, на територіях яких ведуться бойові дії та вони перебувають у максимально нестабільній економічній ситуації, середні ціни складають 2 900 $ та 3 450 $ [13].

1.6. Індекс прозорості ринку вживаних автомобілів України

У 2024 році carVertical зібрали дані у 30 країнах та провели дослідження ринку вживаних автомобілей. «Під час підрахунку відносної вартості пошкоджень ми виключили незначні пошкодження вартістю до 500 євро, такі як зламане бічне дзеркало або невеликі подряпини. Для порівняння ситуації в різних країнах ми поділили середню вартість пошкоджень на ВВП на душу населення», — carVertical. На основі даних було розраховано Індекс прозорості ринку — показник, який вимірює рівень прихованої або неправдивої інформації, яка надається клієнту під час купівлі транспорту. Індекс carVertical базується на 6-факторах з максимальним балом 1,2.

Розрахунок індексу

Для того щоб розрахувати індекс, дані повинні бути стандартизовані — перетворені в єдиний формат, щоб їх можна було порівнювати між різними

змінними. Після стандартизації та призначення ваг змінним, необхідно застосувати просту алгебру для оцінки прозорості країни.

Фактори для обрахунку та їх вага:

- Відсоток одометрів зі скрученим пробігом — 0,3
- Середнє значення скручування одометра — 0,15
- Відсоток пошкоджених автомобілів — 0,2
- Відносна вартість пошкоджень (середня вартість пошкоджень поділена на показник ВВП на душу населення у відповідній країні) — 0,2
- Відсоток імпортованих вживаних автомобілів — 0,3
- Середній вік перевірених автомобілів — 0,05

Кожен фактор впливає по-різному: деякі зменшують прозорість (мають негативний коефіцієнт), інші — збільшують (позитивний коефіцієнт). Чим вищий цей показник, тим прозоріший ринок; чим нижчий — тим більше маніпуляцій і проблем на ринку [14].

Україна займає 25 місце (з результатом — 211,7) серед усіх 30 країн, які брали участь у дослідженні.

Таблиця 1.1

Значення факторів прозорості ринку вживаних автомобілів в Україні

Показник	Значення	Рейтинг
Відсоток одометрів зі скрученим пробігом	9,62 %	24
Середнє значення скручування одометра	99 576 км	23
Відсоток пошкоджених автомобілів	48,82 %	16
Середня вартість пошкоджень	3 485 €	8
Відсоток імпортованих вживаних автомобілів	81,09 %	25
Середній вік перевірених автомобілів	10,47	15

Джерело: побудовано автором на основі [14]

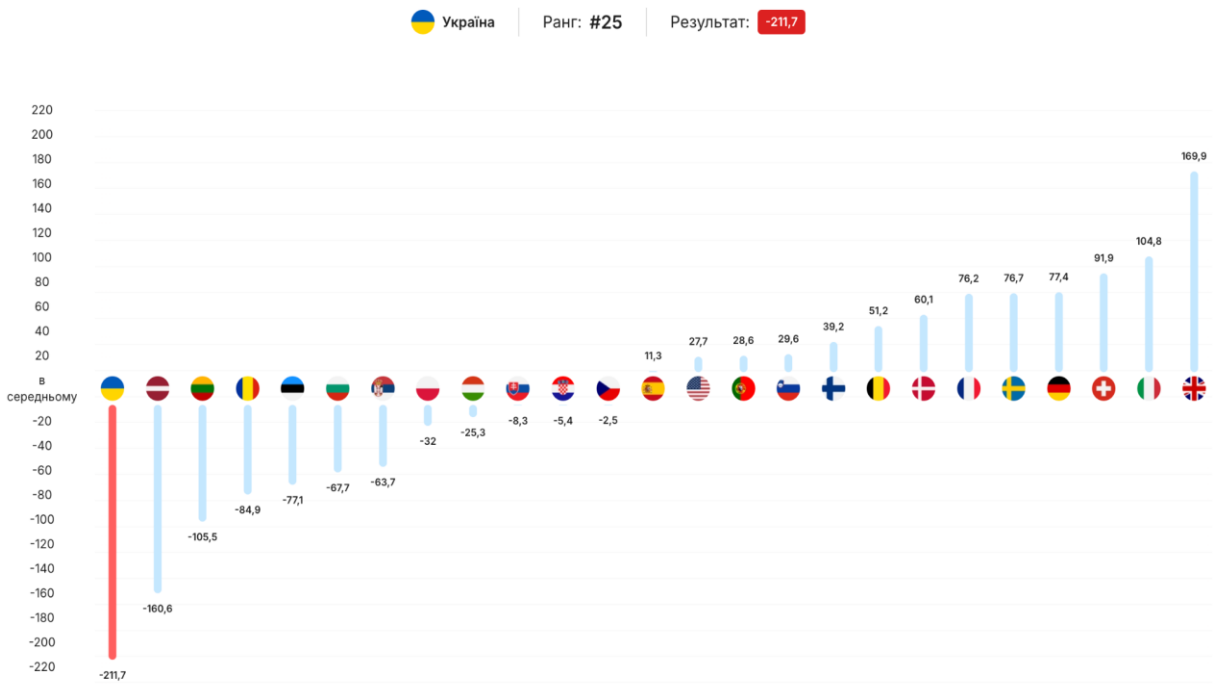

Рис. 1.2. Індекс прозорості ринку вживаних автомобілів України

Джерело: [14]

На рис. 1.2 зображено рейтинги 25 країн. Країнами з високою прозорістю є Великобританія, Італія та Швейцарія, тоді як аутсайдерами є Україна, Латвія та Литва. У 2023 році ситуація була схожою: у трійці лідерів знаходилися Великобританія, Німеччина та Швейцарія.

РОЗДІЛ 2. МЕТОДОЛОГІЧНІ ОСНОВИ ПОБУДОВИ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

2.1. Прогнозна аналітика

Прогнозна аналітика включає в себе використання методів статистики та моделювання для передбачення майбутніх результатів. Для полегшення сприйняття інформації та пошуку залежностей, циклічності або волатильності, прийнято візуалізувати результати у вигляді графіків.

Це може бути корисним для бізнесу або окремих осіб, наприклад:

1. Підприємці можуть прогнозувати майбутні досягнення і на основі цього створювати плани та цілі на майбутнє.
2. Інвестори використовують прогнозування для прийняття рішень під час вкладень коштів.
3. Інтернет-ритейлери використовують прогнозування, щоб точно налаштовувати рекомендації щодо покупок для своїх користувачів і збільшувати продажі.

Під час прогнозування аналітики шукають різні шаблони і циклічності, щоб оцінити ймовірність повторень. Для цього використовуються різні методи прийняття рішень, наприклад: машинне навчання, моделювання, штучний інтелект [15].

Прогнозна аналітика дає можливість зрозуміти побачити майбутнє, дивлячись на події з минулого. **Серед основних методів виділяють:**

1. Машинне навчання — один з методів штучного інтелекту, який використовує різні підходи для створення алгоритмів навчання на основі даних.
2. Прогнозне моделювання — це процес створення математичних моделей, які використовують наявні дані (історичні та поточні), щоб передбачити майбутні події або результати.
3. Добування даних — аналіз великих баз даних, який включає в себе сегментацію та кластеризацію.

4. Історична статистика — дані, які були зібрані в минулому.
5. Поточна статистика — дані, які зібрані в поточному періоді.

Машинне навчання є вужчим поняттям, але тісно пов’язаним з прогнозною аналітикою. Його визначають як підгалузь комп’ютерних наук для навчання та побудови алгоритмів. Часто для моделювання та прогнозування використовують машинне навчання, тобто воно є інструментом прогнозної аналітики [15].

Серед прогнозних моделей виділяють:

- дерева рішень;
- регресію (лінійну та логістичну);
- нейронні мережі;
- кластерні моделі.

Кожну з цих моделей детальніше розглянемо у наступному розділі.

2.2. Моделі для прогнозування

Дерева рішень — це непараметричний метод контрольованого навчання, який застосовується для класифікації та регресії. Основним завданням є створення моделі, яка буде передбачати значення цільової змінної. Навчання відбувається за допомогою простих правил прийняття рішень, які виведені з ознак даних. Уявімо дерево з безліччю гілок і листків. Гілки — доступні варіанти вибору, а листки — конкретне рішення.

Переваги:

- Дерева є простими для розуміння, а також їх легко візуалізувати.
- Не потребують детальної обробки даних і нормалізації.
- Здатні обробляти будь-які дані: числові або категоріальні.
- Можуть обробляти задачі з кількома виходами.
- Легко подаються за допомогою булевої гілки.
- Піддаються статистичній перевірці.

Недоліки:

- Легко перенавчаються.

- Незначні зміни можуть призвести до зміни моделі.
- Роблять передбачення ступінчасто, а не плавно.
- Неможливо побудувати ідеальний прогноз.
- Можуть бути упередженими через переважання даних одного класу.

Отже, дерева рішень є потужним та інтуїтивно зрозумілим інструментом для класифікації та регресії. Їх легко використовувати, вони здатні працювати з різними типами даних, а також потребують лише мінімальну підготовку даних. Дерева підходять для задач, де важлива інтерпретованість моделі та можливість швидкого прогнозування.

Як і кожен метод, дерева рішень не є ідеальними, так як вони схильні до перенавчання, чутливі до змін у даних та не здатні виконувати плавні прогнози для пошуку оптимального рішення. Цей метод підходить для задач, які треба вирішити швидко, тому для побудови складніших моделей краще використовувати інші інструменти [16].

Розглянемо наступний популярний метод у статистиці, який дозволяє знаходити закономірності у великих наборах даних.

Регресійний аналіз — техніка прогнозного моделювання для дослідження зв’язків між цільовою змінною та незалежними. Цей інструмент використовують для прогнозування, моделювання часових рядів, а також знаходження причинно-наслідкового зв’язку між змінними.

Регресію часто використовують у машинному навчанні. Метод досліджує зміну цільової змінної при зміні ознаки та сталості інших факторів. Переваги використання даного методу:

- Виявляє зв’язки між цільовою та незалежними змінними.
- Вказує на силу впливу факторів на залежну змінну.
- Дозволяє порівнювати ефекти змінних, які змінюються у різних масштабах.

При побудові моделі дослідники можуть знайти змінні, які значуще впливають на модель, а також усунути ті, які є зайвими і лише викривлюють результати прогнозу [17].

Наступним методом є **нейронні мережі** — модель, яка імітує людський мозок. Може працювати зі складними взаємозв'язками даних, використовуючи штучний інтелект та розпізнавання шаблонів.

Нейронні мережі складаються з великої кількості шарів вузлів або ж штучних нейронів: вхідного шару, одного або більше прихованих шарів і вихідного шару. Кожен вузол з'єднаний з іншими та має свою власну вагу і порогове значення. Якщо вихід окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані до наступного шару мережі. Інакше, дані не передаються далі до наступного шару мережі.

Для нейронних мереж необхідно мати тренувальні дані, щоб модель могла навчатися та покращувала точність. Цей інструмент дозволяє швидко класифікувати та групувати дані.

Кожен вузол моделі — окрема модель регресії. **Формула наступна**:

$$\sum w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias \tag{2.1}$$

Активаційна функція:

$$output = f(x) = \begin{cases} 1, \sum w_i x_i + b \geq 0 \\ 0, \sum w_i x_i + b < 0 \end{cases}, \tag{2.2}$$

де x_1— вхідні дані, тобто значення, які надходять у нейрон;

w_i — *вага*, яка показує, наскільки важливий відповідний вхід;

$bias$ — зсув.

$f(x)$— активаційна функція, яка вирішує:

- Якщо результат ≥ 0, то вузол активується;
- Якщо < 0, тоді не активується.

Типи нейронних мереж:

1. Перцептрон — одна з перших нейронних мереж, яка здатна лінійно розділяти довільні нелінійні множини.
2. Багатошарові перцептрони — дають можливість будувати складні моделі ділячи різні поверхні і є більш поширеними.
3. Згорткові нейронні мережі — використовуються для розпізнавання зображень за допомогою принципів лінійної алгебри та матричного множення.
4. Рекурентні нейронні мережі — мають внутрішню пам’ять, яка дозволяє пригадувати важливі деталі вхідних даних та з великою точністю прогнозувати майбутні значення [18].

Ще одним важливим інструментом є **кластеризація** — це метод групування даних, на основі схожих характеристик. Популярний інструмент для аналітиків та науковців, які працюють з великими наборами даних. Для побудови моделі вибірку розбивають за схожими ознаками таким чином, щоб кожен кластер був унікальним.

Основною метою метода є передбачення відповідності об’єктів вибірки їхнім класам, щоб сформувати кластери. Таким чином це дає змогу виявити певну новизну, аналізуючи кожен кластер.

Не слід плутати класифікацію та кластеризацію, так як це дещо різні поняття:

Класифікація — метод при якому машина навчається на конкретних прикладах і потім здатна виявляти приналежність нового об’єкту до певного класу.

Кластеризація — інструмент, який групує об’єкти та шукає взаємозв’язки між ними за допомогою алгоритма [19].

2.3. Підготовка до побудови моделі для прогнозування

1. Початковим етапом побудови будь-якої моделі є збір даних. Це крок, на якому аналітик отримує якісні або ж кількісні дані, необхідні для побудови моделі. Наприклад, для прогнозування цін на авто необхідно

зібрати історичні дані про ціни на авто та фактори, які могли на неї впливати.

2. Наступним кроком є обробка даних, яка складається з очистки нерелевантних, невалідних даних та викидів. **Це включає в себе:**

a. видалення дублікатів, пошкоджених і неточних даних;

b. виправлення помилок у наборі даних;

c. уніфікацію форматування даних;

d. виявлення відсутніх даних

3. Третім етапом є розвідувальний аналіз даних, який знаходить початкові найпростіші закономірності, наприклад, розподіл даних, кореляції, мінімальні та максимальні значення тощо.

4. Після цього аналітик використовує методи статистичного аналізу для оцінки даних та формування прогнозів.

a. Описова статистика — розділ статистики, який обробляє емпіричні дані, кількісно представляє результати та візуалізує.

b. Інференційна статистика — розділ статистики, яка робить висновки та прогнози базуючи на зібраних даних [20].

2.4. Розвідувальний аналіз даних

Розвідувальний аналіз даних (Exploratory Data Analysis, EDA) використовується науковцями для аналізу та дослідження наборів даних і узагальнення їх основних характеристик, часто із застосуванням методів візуалізації даних.

Цей метод є важливим кроком у обробці даних, також дозволяє отримати необхідні відповіді, які полегшують виявлення закономірностей, аномалій, припущень або перевірку гіпотез.

Основна мета розвідувального аналізу даних це — з'ясувати, що може показати набір даних, окрім формального моделювання чи перевірки гіпотез, а також краще зрозуміти змінні у даних і взаємозв’язки між ними. Це також

допомагає оцінити доцільність застосування певних статистичних методів для аналізу даних.

Крім того, цей крок дозволяє переглянути дані перед тим, як розбити будь-які припущення. Так легше виявити очевидні помилки, краще зрозуміти закономірності в даних, знайти аномалії або нестандартні події, а також знайти цікаві взаємозв’язки між змінними.

Аналітики часто використовують розвідувальний аналіз для того, щоб переконатися у достовірності та релевантності даних. За допомогою цього інструменту можна знайти відповіді на запитання, пов’язані з середньоквадратичним відхиленням, категоріальними змінними та довірчими інтервалами. Після завершення розвідувального аналізу та отримання інсайтів його результати можуть бути використані для більш складного аналізу або моделювання, включно з машинним навчанням.

За допомогою інструментів розвідувального аналізу даних можна будувати наступні статистичні функції:

- Методи кластеризації та зниження розмірності — допомагають створювати графічні візуалізації багатовимірних даних, що містять велику кількість змінних.
- Інваріативна візуалізація — аналіз кожного окремого поля у «сирому» наборі даних із підрахунком зведеної статистики.
- Варіативна візуалізація та зведена статистика — дозволяє оцінити взаємозв’язок між кожною змінною в наборі даних та цільовою змінною.
- Мультиваріативна візуалізація — для картування та розуміння взаємодій між різними полями в даних.
- Кластеризація методом K-середніх (K-means clustering) — це метод навчання без учителя, при якому точки даних розподіляються на K кластерів залежно від відстані до центроїда кожної групи. Точки, які найближчі до певного центроїда, об'єднуються в один кластер. Метод

широко застосовується в сегментації ринку, розпізнаванні шаблонів та стисненні зображень.

- Прогнозні моделі, як-от лінійна регресія, — використовують статистику та дані для передбачення результатів.

Деякі з найпоширеніших мов програмування в data science, які використовуються для розвідувального аналізу даних (EDA):

- Python — інтерпретована, об’єктно-орієнтована мова програмування з динамічною семантикою. Ця мова вважається високорівневою через вбудовані структури даних, які поєднуються з динамічною типізацією та динамічним зв’язуванням, що і робить Python зручним у використанні. Його бібліотеки дозволяють розробляти додатки, а також використовувати для аналізу даних. Крім того, Python є мовою сценаріїв для інтеграції окремих компонентів. У поєднанні з розвідувальним аналізом даних, це стає потужним інструментом для виявлення відсутніх значень в наборах даних, що є критично важливим для подальшої обробки в задачах машинного навчання.
- R — мова програмування з відкритим кодом, яка має безкоштовне середовище для статистичних обчислень та створення різноманітних графіків. Широко використовується науковцими зі статистики та аналітиками даних для побудови статистичних моделей і проведення аналізу даних.

У своєму досліджені я спиралася лише на мову Python, так як її інструменти дозволяють легко і швидко обробити дані, а також створити візуально привабливі графіки [21].

2.5. Машинне навчання як інструмент для прогнозування

Аналітики часто застосовують алгоритми машинного навчання для аналізу моделей та побудови прогнозних моделей. Для підтримки алгоритму, необхідно постійно давати йому змогу навчатися. Способи навчання:

- Кероване навчання — процес, під час якого моделі окрім сирих вхідних даних дають дані з вже правильними прогнозами. Таким чином модель швидше вчиться розпізнавати закономірності та робити прогнози на нових, схожих даних.
- Некероване навчання — процес, який передбачає повністю самостійне навчання моделі. Тобто модель не має правильних відповідей і їй самій треба шукати закономірності [20].

Здебільшого для аналізу аналітики розподіляють дані на базову та тестову групи. Навчальні дані мають відомі вхідні та відповідні вихідні значення, що навчають алгоритми робити точні прогнози. Тестові дані дозволяють професіоналам оцінювати ефективність моделі [20].

Для того, щоб оцінити ефективність моделей прогнозування та алгоритмів машинного навчання, можна порівнювати прогнозовані значення з реальними даними. Для оцінки точності моделей часто використовують наступні метрики:

- Середньоквадратичну помилку для порівняння передбачених значень з реальними.
- Середню абсолютну помилку для оцінки різниці між набором передбачених значень та реальних.
- Критерій коефіцієнта варіації для ідентифікації патернів за допомогою обчислення відстані між точками даних різних кластерів [20].

Важливим етапом є вибір ознак — процес визначення змінних, які впливають на результати моделі. Цей етап допомагає ідентифікувати змінні, які покращують прогнози моделі.

Останнім кроком є інтерпретування ознак. Процес передбачає аналіз результатів моделі. Після цього необхідно представити прогнози у вигляді діаграграм та графік для встановлення зв’язків між даними.

2.6. Моделі для прогнозування

Алгоритм випадкового лісу — це потужний спосіб навчання на деревах у машинному навчанні для формування різних передбачень. Метод побудований

таким чином, щоб провести голосування всіх дерев для отримання кінцевого передбачення. Він широко застосовується для задач класифікації та наближення.

Даний алгоритм є різновидом класифікатора, який використовує багато дерев для формування рішень. Спочатку беруться випадкові частини набору даних для навчання кожного дерева, а потім вони об’єднуються у результати шляхом обчислення середнього значення або більшості значень. Такий підхід підвищує точність передбачень. Випадковий ліс ґрунтується на спільному навчанні.

Першим кроком у побудові дерева рішень є отримання набору даних, який містить рядки та відповідні до них ознаки — стовпці.

Наступним кроком є створення багатьох дерев рішень на основі навчального набору даних. Кожне дерево навчається на випадковій підмножині даних і випадковому наборі ознак. Цей процес називається мішкуванням або об’єднанням з поверненням.

Кожне дерево рішень самостійно, незалежно від інших, формує передбачення.

Коли надходить новий, раніше не бачений приклад, кожне дерево ще раз запускає процес формування рішення.

Остаточне передбачення формується шляхом об’єднання передбачень усіх дерев. Для класифікації це відбувається через голосування більшістю, а для наближення — через обчислення середнього.

Основні властивості випадкового лісу:

- Обробка пропущених значень. Дерева автоматично працюють з відсутніми значеннями під час навчання, що усуває потребу у мануальному заповненні.

- Оцінка важливості ознак. Метод визначає найважливіші ознаки, які чинять найбільший вплив на прогнозування. Це допомагає у відборі ознак і поясненні результатів.
- Робота з великими та складними даними. Легко обробляє великі масиви даних без втрат продуктивності.
- Універсальність. Може застосовуватися як для класифікації, так і для наближення.
- Випадковість. При створенні дерева випадковим чином обирається підмножина ознак для розділення даних, а не використовуються всі наявні ознаки одразу. Це дозволяє уникнути перенавчання та робить передбачення точнішими та більш надійними.

Припущення алгоритму випадкового лісу:

1. Кожне дерево в лісі робить своє передбачення, не покладаючись на інші.
2. Дерева будуються на основі випадкових вибірок і ознак, щоб зменшити ймовірність помилок.
3. Великі обсяги даних забезпечують різноманіття дерев, що дозволяє їм вивчити унікальні шаблони.
4. Об’єднання передбачень різних дерев дає точніший підсумковий результат [22].

Ще однією моделлю є **градієнтне підсилення** — спосіб об'єднаного навчання, який застосовують для класифікаційних і регресійних завдань. Цей метод базується на поєднанні кількох слабких навчальних моделей для створення сильної моделі для прогнозування. Градієнтне підсилення послідовно навчає моделі, де кожна наступна виправляє помилки попередньої.

У цьому методі кожна нова модель навчається з метою зменшення функції втрат (наприклад, середньоквадратичної похибки або перехресної ентропії), використовуючи метод спуску за градієнтом. Кожен крок дає алгоритму змогу

обчислювати градієнт функції втрат відносно передбачень, а потім навчає нову слабку модель, щоб зменшити цей градієнт. Передбачення нової моделі додаються до загального результату, і процес повторюється доти, доки не досягеться критерій зупинки.

Ключовою властивістю градієнтного підсилення є зменшення внеску, що масштабує вплив кожної нової моделі за допомогою коефіцієнта швидкості навчання (позначається як η), де:

- Менші значення швидкості навчання означають, що внесок кожного дерева є меншим, що зменшує ризик перенавчання, але потребує більшої кількості дерев для досягнення такої ж точності.
- Більші значення швидкості навчання означають, що кожне дерево має більший вплив, але це може призвести до перенавчання.

Існує баланс між швидкістю навчання і кількістю оцінювачів (дерев): менша швидкість зазвичай вимагає більшої кількості дерев для досягнення найкращих результатів.

Особливості градієнтного підсилення:

1. Послідовне навчання. Створюється об’єднання багатьох дерев, кожне з яких навчається виправляти помилки попереднього. Спочатку перше дерево навчається на початкових даних *x* і правильних відповідях *y*. Воно створює передбачення, які використовуються для обчислення залишків (різниці між справжніми та передбаченими значеннями).
2. Обчислення залишків. Другий крок складається з тренування другого дерева, яке навчається на ознаках *x* і залишках з першого дерева як мітках. Тобто друге дерево вчиться передбачати помилки першого. Далі дані передаються до наступного дерева і процес знову повторюється доти, доки не пройде увесь ліс.
3. Зменшення внеску. Після завершення навчання кожного дерева, його результат зменшується множенням на швидкість навчання η

(значення від 0 до 1). Це знижує ризик перенавчання, обмежуючи вплив кожного дерева на підсумкову модель.

Коли навчання завершилося, передбачення обчислюються шляхом підсумовування внесків усіх дерев. Остаточне передбачення визначається за формулою:

$$y_{predicted} = y_1 + \eta \cdot r_1 + \eta \cdot r_2 + ... + \eta \cdot r_n, \quad (2.3)$$

де r_n — це залишки (помилки), передбачені кожним деревом [23].

2.7. Бібліотеки та популярні модулі у машинному навчанні

Для швидкого збору відкритих даних з сайтів, можна використовувати бібліотеку **Requests**. Це стандартний інструмент для створення HTTP-запитів у Python. Вона абстрагує складність роботи з HTTP і надає простий та зручний API, що дозволяє зосередитися на взаємодії з сервісами та обробці даних.

Переваги бібліотеки Requests:

- Проста і синтаксично зрозуміла.
- Дозволяє надсилати HTTP-запити до сайтів та API.
- Дає змогу легко обробляти відповіді сервера.
- Гнучко налаштовується для різних ситуацій.

Бібліотека Requests не підтримує асинхронні HTTP-запити напряму. Для цього необхідно використовувати AIOHTTP або HTTPX, яка має схожий синтаксис до Requests.

Основними можливостями цієї бібліотеки є:

1. GET-запит, який здебільшого використовується для отримання даних або ж доступу до ресурсу. Код для використання:

```
import requests
response = requests.get('https://example.com')
```

Як можна було помітити у коді, дані у GET-запитах передаються через URL.

2. POST-запит, який використовується для надсилання даних, здебільшого для створення або ж оновлення ресурсу. Код виглядає наступним чином:

```
import requests
requests.post("https://example.com/post", data={"key": "value"}) [24]
```

Таблиця 2.1

Порівняння запитів Get і Post

Характеристика	GET	POST
Використання	Отримання ресурсу	Створення або оновлення ресурсу
Формат передачі даних	У параметрах URL	У тілі запиту
Безпечність	Менш безпечний (дані видно в URL)	Більш безпечний (дані приховані)
Обмеження довжини	Є обмеження URL	Немає обмежень на розмір даних
Можна зберегти в URL?	Так	Ні

Джерело: побудовано автором на основі [24]

Бібліотека Requests є чудовим інструментом для HTTP-запитів у Python, особливо для тих, хто хоче просто і швидко взаємодіяти з API або сайтами.

Beautiful Soup (bs4) — це бібліотека Python для витягування даних з файлів у форматах HTML та XML. Вона працює разом з парсером для зручної навігації та модифікації дерева розбору. Ця бібліотека допомагає швидко здійснити парсинг. Код виглядає наступним чином:

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
```

Beautiful Soup перетворює HTML документ на складне дерево об'єктів Python.

Для роботи з HTML документами бібліотека визначає підкласи NavigableString, які витягує рядки з HTML тегів. Таким чином легше виділяти рядки потрібного тексту сторінки з ігноруванням рядків, які є директивами програмування.

Для обробки XML файлів Beautiful Soup визначає деякі класи NavigableString для зберігання спеціальних типів рядків, які можна знайти в XML-документах. Як і Comment, ці класи є підкласами NavigableString, які надають додаткову інформацію до рядка при виведенні.

Beautiful Soup є зручною бібліотекою для парсингу HTML і XML у Python. За допомогою неї простіше витягувати дані, а також вона працює з кількома типами об’єктів та значно економить час при обробці веб-сторінок [25].

Pandas — це програмна бібліотека мови Python, яка надає пакет для обробки та аналізу даних. Вона пропонує структури даних і операції для роботи з числовими таблицями та часовими рядами [26].

Бібліотека насичена різними функціями для аналізу, очищення, дослідження та обробки даних. За допомогою цієї бібліотеки легко аналізувати великі обсяги даних, а також оцінювати результати на основі статистичних теорій.

Pandas надає інструменти для очистки неструктурованих наборів даних та викидів, а також перетворює їх на читабельні та релевантні. За допомогою інструментів цієї бібліотеки також можна відповісти на питання:

- Чи є кореляція між двома або більше стовпцями?
- Яке середнє значення у наборі даних?
- Яке максимальне значення?
- Яке мінімальне значення?

Pandas містить багато функції, які раніше були доступними лише у мові програмування R. Тепер бібліотека містить багато інструментів для роботи з

DataFrame, дає змогу легко видалити нерелевантні рядки або ж порожні значення, а також має широкий функціонал для інших операцій з даними [27].

NumPy — одна з найпопулярніших бібліотек мови Python, яку використовують для роботи з масивами даних. За допомогою неї можна вирушувати задачі лінійної алгебри, перетворення Фур'є та працювати з матрицями.

Ця бібліотека була створена у 2005 році з відкритим вихідним кодом, який можна вільно використовувати. Назва NumPy є скороченою від Numerical Python — чисельний Python.

У Python існують списки, які можуть виконувати функції масивів. Однак їх обробка є досить повільною, особливо при роботі з великими обсягами даних.

NumPy забезпечує об’єкти масиву, який працює до 50 разів швидше за звичайні списки Python. Масиви є надзвичайно важливими у роботі з даними, і саме згадана бібліотека має безліч функцій, які значно спрощують обробку цих масивів. NumPy дозволяє швидко та ефективно обробити їх.

Кожен масив зберігається в пам’яті одним безперервним блоком, тоді як списки Python зберігаються у фрагментованому вигляді. Таким чином процесор може швидше отримувати доступ до даних і ефективно ними оперувати.

Така особливість називається локальність посилань у комп’ютерних науках. Саме вона забезпечує високу продуктивність NumPy порівнняно зі списками. До того ж, NumPy оптимізований для роботи з сучасними архітектурами процесорів [28].

SciPy — це безкоштовна бібліотека з відкритим кодом для мови Python, яка використовується для наукових і технічних обчислень.

Вона складається з модулів для оптимізації, лінійної алгебри, інтегрування, інтерполяції, спеціальних функцій, швидкого перетворення Фур'є, обробки сигналів та зображень, розв’язування звичайних диференціальних рівнянь та інших задач, які є поширеними у науці та інженерії.

SciPy є основою для наукових обчислень у Python. **До складу пакета входять такі підпакети:**

- cluster — ієрархічна кластеризація, векторна квантизація, K-середніх;
- constants — фізичні константи та коефіцієнти перетворення;
- fft — алгоритми для дискретного перетворення Фур'є;
- integrate — безліч методів інтегрування;
- interpolate — інструменти для інтерполяції;
- io — введення та виведення даних;
- linalg — функції для лінійної алгебри;
- misc — різні допоміжні утиліти;
- ndimage — функції для обробки багатовимірних зображень;
- ODR — алгоритми та класи для ортогональної регресії;
- optimize — алгоритми оптимізації, включаючи лінійне програмування;
- signal — інструменти для обробки сигналів;
- sparse — розріджені матриці та пов’язані алгоритми;
- spatial — алгоритми для просторових структур (k-d дерева, найближчі сусіди, опуклі оболонки тощо);
- special — спеціальні функції;
- stats — статистичні функції;
- weave — інструмент для написання коду на C/C++ у вигляді багаторядкових рядків Python.

Найголовнішою структурою даних цієї бібліотеки є багатовимірний масив, який надає бібліотека NumPy. Попри те, що NumPy також має багато функцій, SciPy має перевагу, коли необхідно працювати з науковими обчисленнями. Все ж, ці дві бібліотеки чудово поєднувати, коли необхідно обробити великі масиви та інтегрувати різноманітні бази даних [29].

У Python бібліотека є ще бібліотека **statsmodels**, яка використовується для оцінки статистичних моделей і проведення статистичних тестів. Вона побудована на основі numpy, scipy та pandas.

Бібліотека широко використовується в економетриці, а також у таких галузях, як фінанси, маркетинг і соціальні науки. Вона підтримує різні моделі, зокрема лінійну регресію, узагальнені лінійні моделі, аналіз часових рядів тощо.

Серед основних можливостей виділяють наступні:

- Оцінка статистичних моделей. Бібліотека має багато різних класів та функцій, які здатні оцінювати різні статистичні моделі, такі як лінійна регресія, узагальнені лінійні моделі, аналіз часових рядів тощо.
- Статистика. За допомогою Statsmodels можна проводити статистичні тести та перевіряти різні гіпотези.
- Аналіз. У бібліотеки є безліч функцій, за допомогою яких можна проводити дослідження та аналізувати дані, зокрема з описової статистики, аналізу кореляцій тощо.
- Візуалізація. Statsmodels багата функціями для візуалізації даних, наприклад: діаграми розсіювання, гістограми, стовпчикові діаграми тощо.
- Інтеграція з іншими бібліотеками. Так як бібліотека побудована на основі numpy, scipy і pandas, вона легко інтегрується з іншими бібліотеками екосистеми Python [30].

Scikit-learn — це популярна відкрита бібліотека для машинного навчання, написана мовою Python. За допомогою неї легко реалізовувати моделі штучного інтелекту та статистичного моделювання завдяки уніфікованому інтерфейсу. Бібліотека побудована на основі NumPy, SciPy та Matplotlib.

У ній доступні алгоритми для:

- класифікації;
- регресії;

- кластеризації;
- зниження розмірності.

Бібліотека дозволяє швидко реалізовувати як навчання з учителем (класифікація, регресія), так і без учителя (кластеризація, зниження розмірності), не вдаючись у складні математичні деталі.

Компонентами scikit-learn є:

- NumPy, яка надає ефективні масиви та операції з ними, важливі для обробки даних.
- SciPy, яка є розширення NumPy і орієнтована на наукові обчислення.
- Matplotlib, яка є гнучким інструментом для створення графіків та візуалізації даних.
- Cython, яка дозволяє оптимізувати виконання Python-коду за допомогою компіляції в C.

Scikit-learn також надає різноманітні інструменти для підготовки даних до машинного навчання, тобто попередньої обробки:

1. Масштабування числових ознак:
   - StandardScaler – стандартизує ознаки;
   - MinMaxScaler – нормалізує значення в заданий діапазон.
2. Кодування категоріальних змінних:
   - OneHotEncoder – створює окрему бінарну ознаку для кожної категорії;
   - LabelEncoder – замінює категорії числовими мітками.
3. Обробка пропущених значень:
   - SimpleImputer – заповнює порожні значення середнім, медіаною тощо.
4. Вибір ознак:
   - RFE – поступове видалення неважливих ознак;

- mutual_info_classif / mutual_info_regression – визначає взаємозв’язок між ознаками та цільовою змінною.

Scikit-learn дозволяє поєднувати ці методи в пайплайни, що забезпечує автоматизовану та послідовну обробку даних.

Для оцінки якості моделей застосовують такі методи як частка правильних передбачень, точність позитивних передбачень, повнота, тобто здатність знаходити всі позитивні приклади, середнє гармонійне точності та повноти. Крім того, можна побудувати графік, який відображатиме баланс між True Positive Rate та False Positive Rate – AUC-ROC.

Для оцінки якості регресійної моделі використовують середнє абсолютне відхилення, середньоквадратичну помилку та коефіцієнт детермінації [31].

Matplotlib — універсальна бібліотека Python, у якої відкритий код. Її використовують для зручної візуалізації даних у різних форматах. Функціонал дозволяє графічно представити інформацію, що значно полегшує аналіз і розуміння даних.

У Matplotlib є багато інструментів для створення простих лінійних графіків, які найчастіше використовують у аналітиці. Базові графіки генеруються кількома рядками коду, що робить бібліотеку дуже зручною для початківців. Кожен графік легко виводиться на екран, що дає змогу відразу бачити результат.

До основних елементів графіків належать:

- *Figure* — головний контейнер для всіх елементів графіка, полотно, де знаходяться всі елементи.
- *Axes* — ділянки всередині фігури, де безпосередньо відображаються дані.
- *Axis* — осі X та Y, які мають шкали, підписи та межі.
- *Lines і Markers* — лінії показують зв’язки між точками, а маркери виділяють окремі значення.
- *Title і Labels* — заголовки та підписи осей, які дають змогу краще зрозуміти, що саме зображено на графіку.

Окремим модулем бібліотеки є Pyplot, інтерфейс якого подібний до MATLAB, що спрощує побудову графіків. Інструментарій дає змогу додавати до графіка різні елементи, наприклад: лінії, текст, зображення тощо. Основні етапи роботи включають імпорт модуля, створення даних, побудову графіка, його налаштування (заголовки, підписи осей) і фінальний виклик функції для відображення.

Matplotlib дозволяє побудувати різноманітні типи графік, що дає змогу адаптувати візуалізацію до різних потреб. **Найпопулярніші типи включають:**

1. Лінійні графіки
2. Стовпчикові діаграми
3. Гістограми
4. Діаграми розсіювання
5. Кругові діаграми
6. 3D-графіки

Основними перевагами Matplotlib є те, що ця бібліотека гнучка та багатофункціональна. За допомогою неї можна створювати різні графіки: як прості двовимірні, такі і складні, багатовимірні візуалізації. Користувач може повністю налаштувати графік, використовуючи різні кольори, шрифти, стилі. Є можливість самостійно розташовувати підписи та встановлювати розміри елементів.

Бібліотеку легко інтегрувати з NumPy, щоб напряму будувати графіки на основі різних масивів даних. Усі графіки мають високу якість, тому за потреби їх можна публікувати без хвилювань за читабельність. Крім того, цю бібліотеку підтримують усі наявні операційні системи. Графіки можна використовувати у динамічному режимі, що робить бібліотеку ще зручнішою для аналізу.

Matplotlib найчастіше використовують для візуалізації даних у Python, особливо для побудови статичних, анімованих та інтерактивних графіків. Вона є чудовим інструментом у наукових дослідженнях, аналітиці, машинному навчанні та візуальному представленні статистики [32].

Ключові напрямки використання Matplotlib включають:

- Створення базових графіків (лінії, стовпчики, гістограми, розсіювання).
- Візуалізацію статистичних показників (діаграми з помилками, boxplots).
- Гнучке налаштування стилю та вигляду графіка.
- Побудову кількох графіків в одній фігурі.
- Побудову 3D-графіків.
- Створення анімацій та інтерактивних візуалізацій.
- Інтеграцію з Pandas, NumPy і Jupyter Notebook.

Ще однією популярною бібліотекою для візуалізації є **seaborn**. У мові програмування Python існує чимало бібліотек для побудови графіків, зокрема вже добре відома matplotlib. Така різноманітність бібліотек перш за все пов’язана з тим, що вдала візуалізація є однією з найважливіших частин аналітики та сторітелінгу, адже саме вона дозволяє швидко виявляти важливу інформацію. Кожен інструмент має як переваги, так і недоліки, тому обізнаність у тому, що існує багато бібліотек для виконання певних завдань допомагає науковцям обрати ті, які найкраще відповідають поставленим цілям.

Seaborn є потужною бібліотекою для побудови статистичних графіків у Python. Це ще одна бібліотека, яка має широкий функціонал для створення візуально привабливих графіків з гарними стилями та палітрами кольорів за замовчуванням.

Бібліотека побудована на базі matplotlib, але тісно інтегрується з об’єктами з pandas, що робить її зручною для роботи з таблицями даних. Основною метою seaborn є зробити візуалізацію центральним інструментом для дослідження та розуміння даних. Вона пропонує API, який орієнтований на датасети, що дозволяє легко перемикатися між різними типами графіків для одного й того ж набору змінних.

У seaborn є багато типів графіків, які можна поділити на кілька наступних категорій:

- Реляційні графіки — використовуються для аналізу зв’язків між двома змінними.
- Категоріальні графіки — призначені для візуалізації змінних, що представляють категорії (наприклад, групи або класи).
- Графіки розподілу — дозволяють аналізувати розподіл даних за однією або двома змінними.
- Регресійні графіки — використовуються для виявлення трендів і шаблонів, які можуть бути корисними під час дослідження даних.
- Матричні графіки — є масивом точкових графіків і дозволяють бачити взаємозв’язки між багатьма змінними одночасно.
- Графіки у сітці — дають змогу побудувати кілька версій одного й того ж графіка для різних підгруп даних.

Основними типами графіків у Seaborn є:

- Гістограма, яка використовується для візуалізації розподілу даних за однією змінною. Можна налаштувати кількість інтервалів та додавати графіки щільності. Це один з найпростіших способів побачити форму розподілу.
- Діаграма розподілу — подібна до гістограми, але комбінує одразу кілька графіків: гістограму, графік щільності та ковдру.
- Лінійний графік — найбільш базовий графік, який відображає зміну значень змінної у часі. Особливо корисний для візуалізації тимчасових рядів або послідовностей даних.

За допомогою цих інструментів можна зручно аналізувати і представляти дані. Через наявність гнучкого функціоналу можна підвищити розуміння структури, розподілу та зв’язків у наборі даних. Seaborn є важливим інструментом для аналітиків на всіх рівнях [33].

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ЦІН НА ВЖИВАНІ АВТОМОБІЛІ

3.1. Підготовка до моделювання

Український ринок автомобілів зазнав значного впливу повномасштабного вторгнення: якщо колись продавці як публікували оголошення, так і продавали транспортні засоби на офлайн ринках, то тепер значна перевага надається онлайн продажам.

Найдорожчі автомобілі продають у Києві, Київській, Львівській, Волинській, Чернівецькій та Одеській областях. Найдешевші авто знаходяться у Херсонській, Донецькій та Луганській областях, що спричинено частковою окупацією або активними бойовими діями

Метою створення моделі було зрозуміти залежності між ціною вживаного автомобіля та різними факторами. Для цього необхідно було зібрати дані з якоїсь платформи продажу автомобілів, щоб модель навчалася на реальних даних і могла в майбутньому прогнозувати реальну ціну на автомобіль, який був у використанні [34].

У своєму аналізі я створювала модель на основі популярного сайту auto.ria [6]. Щоб отримати дані про автомобілі, спочатку треба було написати код, який обробив основну інформацію на сайті та зберіг їх у базі даних з назвою, ціною та характеристикою кожного автомобіля.

Для парсингу необхідно було імпортувати бібліотеки та модулі, які детально розглядалися у минулому розділі:

Бібліотеки [35]:

1. Requests — бібліотека, яка використовується для надсилання HTTP-запитів. За допомогою неї легко отримувати дані з вебсайтів (GET, POST тощо).
2. bs4 — інструмент для парсингу HTML та XML. Дозволяє витягати потрібну інформацію зі сторінки.

3. Pandas — бібліотека для роботи з табличними даними. Дозволяє зчитувати, обробляти, аналізувати й візуалізувати дані.

Модулі [35]:

1. csv — модуль для читання та запису CSV-файлів.
2. Time — модуль для роботи з часом. Містить функції для затримок, наприклад, sleep, яка особливо корисна для парсингу, щоб зменшити навантаження на ресурс.
3. re — модуль для роботи з регулярними виразами. Дозволяє здійснювати пошук, заміну та перевірку шаблонів у тексті.
4. Datetime — вбудований модуль для роботи з датами і часом. Використовується для обчислень з датами, форматування часу тощо.

Основна функція створювалася за допомогою наступного коду:

```
for page in range(1, NUM_PAGES + 1):
    print(f"Збираємо дані з сторінки {page}...")
    url = f"{BASE_URL}?page={page}"
    response = requests.get(url, headers=HEADERS)
    response.raise_for_status()
    soup = BeautifulSoup(response.text, 'html.parser')
    car_cards = soup.find_all('section', class_='ticket-item')

    for car_card in car_cards:
        car_data = {
            "brand": None,
            "year": None,
            "selling_price": None,
            "km_driven": None,
            "city": None,
            "publication_date": None,
```

```
    "Engine (CC)": None,
    "fuel": None,
    "transmission": None
}
```

Виглядав файл з даними наступним чином:

	brand	year	selling_price	km_driven	city	publication_date	Engine (CC)	fuel	transmission
1	Audi Q7	2013	22000	154	Харків	2025-04-17	3	Дизель	
2	Audi Q7	2018	34900	89	Стрий	2025-04-17	3	Бензин	Автомат
3	Mercedes-Benz Sprinter	2020	39500	136	Рівне	2025-04-16	2.2	Дизель	Автомат
4	Tesla Model 3	2023	29500	28	Біла Церква	2025-04-17			Автомат
5	Tesla Model 3	2018	19999	143	Одеса	2025-04-15			
6	BMW X3	2021	39950	42	Івано-Франківськ	2025-04-02	2	Бензин	Автомат
7	BMW X3	2020	40999	159	Луцьк	2025-04-17	3	Дизель	Автомат
8	Volkswagen Tiguan	2020	25500	57	Київ	2025-04-11	2	Бензин	Автомат
9	Toyota Camry	2013	16700	194	Київ	2025-04-12	2.5	Бензин	Автомат
10	Audi Q7	2016	27900	250	Виноградів	2025-03-23	3	Бензин	Автомат
11	BMW 5 Series	2017	34000	191	Львів	2025-04-13	2	Дизель	Автомат
12	Mercedes-Benz Sprinter	2019	30900	263	Рівне	2025-04-16	2.2	Дизель	Ручна / Механіка

Рис. 3.1. Датасет

Джерело: побудовано автором у Jupyter Notebook

У ньому зібрані такі дані: бренд автомобіля, рік випуску, ціна продажу (у доларах США), пробіг, місто продажу, дата публікації, об’єм двигуна, паливо та тип трансмісії. Це основні фактори, які характеризують автомобіль.

У файлі було зібрано 4 000 оголошень з сайту, які знаходяться на перших 200 сторінках. Деякі рядки мають незаповнені колонки, які будуть викривлювати результати моделі, тому під час аналізу їх необхідно було ідентифікувати та видалити.

3.2. Розвідувальний аналіз даних

Наступний крок полягав у проведенні розвідувального аналізу даних, який дозволяє оцінити набори даних та виокремити їхні властивості та візуалізувати результати.

Для цього додатково імпортуємо такі бібліотеки як [35]:

1. **Numpy** — бібліотека для наукових обчислень та роботи з масивами, математичними операціями, статистикою.
2. **Scipy** — бібліотека для наукових і статистичних розрахунків.

3. **Statsmodels** — бібліотека для статистичного моделювання та економетрики.

4. **Sklearn** — бібліотека для машинного навчання, підготовки даних, нормалізації тощо.

5. **Matplotlib** – бібліотека для створення статичних, анімованих та інтерактивних графіків у Python. Вона є однією з найпопулярніших для візуалізації даних.

6. **seaborn** – бібліотека, яка робить графіки візуально привабливими та надає багато інструментів для аналізу залежностей у даних. Містить функції для побудови різних типів статистичних графіків.

7. **scikit-learn** – бібліотека, яка містить інструменти для класифікації, регресії, кластеризації, зниження розмірності, попередньої обробки даних тощо.

8. **statsmodels** – бібліотека для статистичного аналізу та побудови моделей, зокрема регресійного аналізу, аналізу дисперсії (ANOVA), часових рядів, тестування гіпотез.

Розглянемо кроки для детальної обробки даних. Одним з них є виявлення пропущені значень та їх видалення, щоб вони не викривлювали результати:

```
df = df.dropna(axis=0, how='any')
```

Наступним кроком обробляємо дані:

1. Конвертуємо дати у числовий формат:

```
df_prepared['year'] = pd.to_datetime(df_prepared['year'], format='%Y').dt.year
df_prepared['publication_date'] = pd.to_datetime(df_prepared['publication_date']).dt.month
```

2. Обробляємо категоріальні змінні:

```
categorical_columns = ['brand', 'city', 'fuel', 'transmission']
label_encoders = {}
for col in categorical_columns:
label_encoders[col] = LabelEncoder()
```

```
df_prepared[col] = label_encoders[col].fit_transform(df_prepared[col])
```

3. Нормалізуємо числові змінні:

```
numeric_columns = ['year', 'km_driven', 'Engine (CC)', 'publication_date']
scaler = StandardScaler()
df_prepared[numeric_columns] = scaler.fit_transform(df_prepared[numeric_columns])
```

4. Розділяємо дані на незалежні (x) та залежну змінну (y):

```
x = df_prepared.drop('selling_price', axis=1)
y = df_prepared['selling_price']
```

Після того, як дані мінімально підготовлені до аналізу, треба провести аналіз статистичної значущості, щоб виявити фактори, які дійсно впливають на ціну, та такі, які не є значущими.

1. Базова статистика:

```
results = {
    'model_summary': model.summary(),
    'r_squared': model.rsquared,
    'adj_r_squared': model.rsquared_adj,
    'f_statistic': model.fvalue,
    'f_pvalue': model.f_pvalue
}
```

```
Базова статистика моделі:
R²: 0.5019
Adjusted R²: 0.5004
F-статистика: 323.0935
P-значення F-тесту: 0.0000
Якість моделі:
- R-квадрат: 0.502
- Скоригований R-квадрат: 0.500

Модель є статистично значущою (p < 0.05)
```

Рис. 3.2. Базова статистика моделі

Джерело: побудовано автором у Python

2. Проводжу аналіз значущості кожного фактору:

```
feature_significance = pd.DataFrame({
    'coefficient': model.params,
    'std_err': model.bse,
    't_value': model.tvalues,
    'p_value': model.pvalues,
    'significance': ['Significant' if p < 0.05 else 'Not significant' for p in model.pvalues]
})
```

	coefficient	std_err	t_value	p_value	significance
const	22,274.8264	961.0941	23.1765	0.0000	Significant
brand	-6.0829	2.6695	-2.2787	0.0228	Significant
year	8,045.9504	415.1595	19.3804	0.0000	Significant
km_driven	-10,588.0419	485.8851	-21.7912	0.0000	Significant
city	-18.5509	6.0998	-3.0412	0.0024	Significant
publication_date	505.8410	334.4595	1.5124	0.1306	Not significant
Engine (CC)	12,081.0632	424.5260	28.4578	0.0000	Significant
fuel	1,743.6555	196.9364	8.8539	0.0000	Significant
transmission	986.8424	885.6557	1.1143	0.2653	Not significant

Рис. 3.3. Статистистична значимість коефіцієнтів

Джерело: побудовано автором у Python

Статистично значущими факторами впливу на цілу виявилися:

- Бренд з негативним впливом (коеф. = -6.083, p = 0.0228).
- Рік з позитивним впливом (коеф. = 8045.950, p = 0.0000).
- Пробіг з негативним впливом (коеф. = -10588.042, p = 0.0000).
- Місто з негативним впливом (коеф. = -18.551, p = 0.0024).
- Об’єм двигуна з позитивним впливом (коеф. = 12081.063, p = 0.0000).
- Паливо з позитивним впливом коеф. = 1743.656, p = 0.0000).

Поки якість моделі є не дуже високою: R-квадрат = 0.502, скоригований R-квадрат = 0.500 через вплив незначущих факторів.

Щоб модель краще працювала, створюємо додаткові змінні:

- місяць публікації;
- ціна за рік віку (вартість амортизації);
- ціна за кубічний сантиметр двигуна;
- категорія віку авто;
- категорія пробіг;
- середній річний пробіг;
- сезон публікації;
- класифікація авто за ціною.

Розглянемо оброблені результати аналізу:

Найбільше автомобілів знаходиться у ціновому діапазоні 9 000$ — 15 000$, далі з підвищенням ціни кількість автомобілів зменшується.

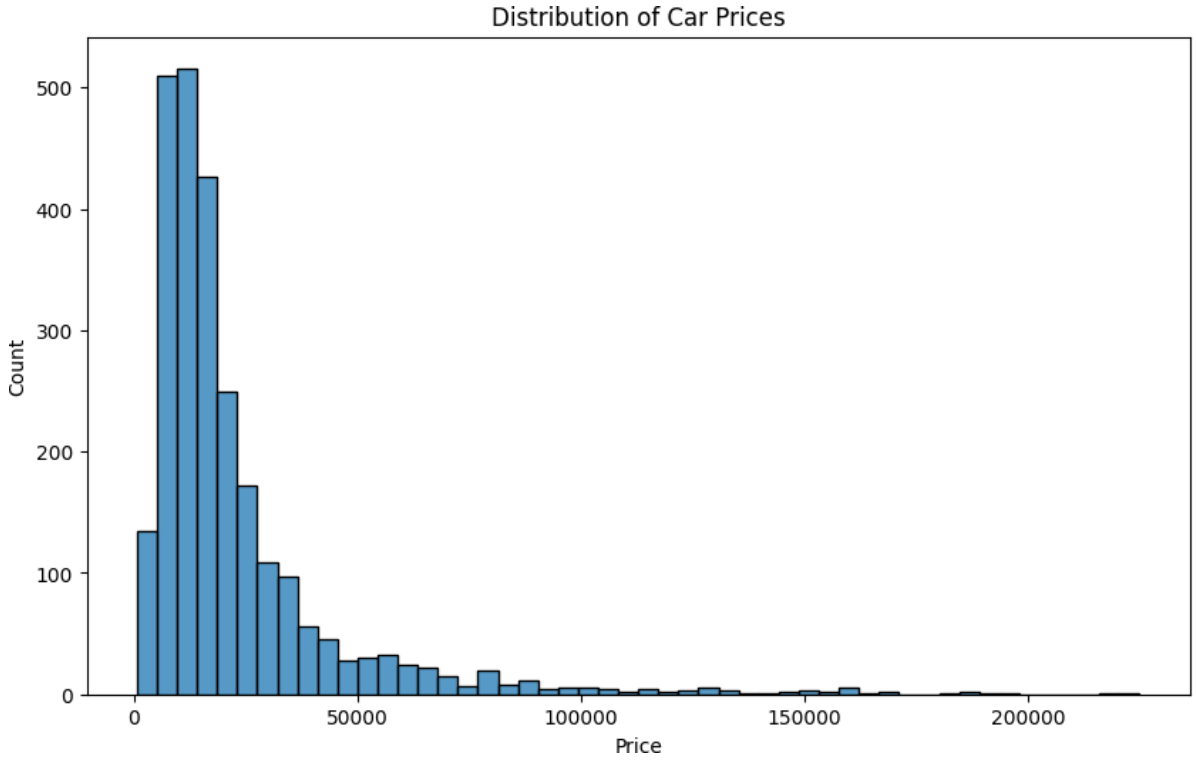

Рис. 3.4. Ціновий розподіл автомобілів

Джерело: побудовано автором у Python

Ціна також залежить від типу палива. Найдорожчі автомобілі мають гібридний тип палива, а найдешевші — газ. Найбільше викидів спостерігається для автомобілів, які їздять на дизелі або бензині, що видно на рис. 3.4.

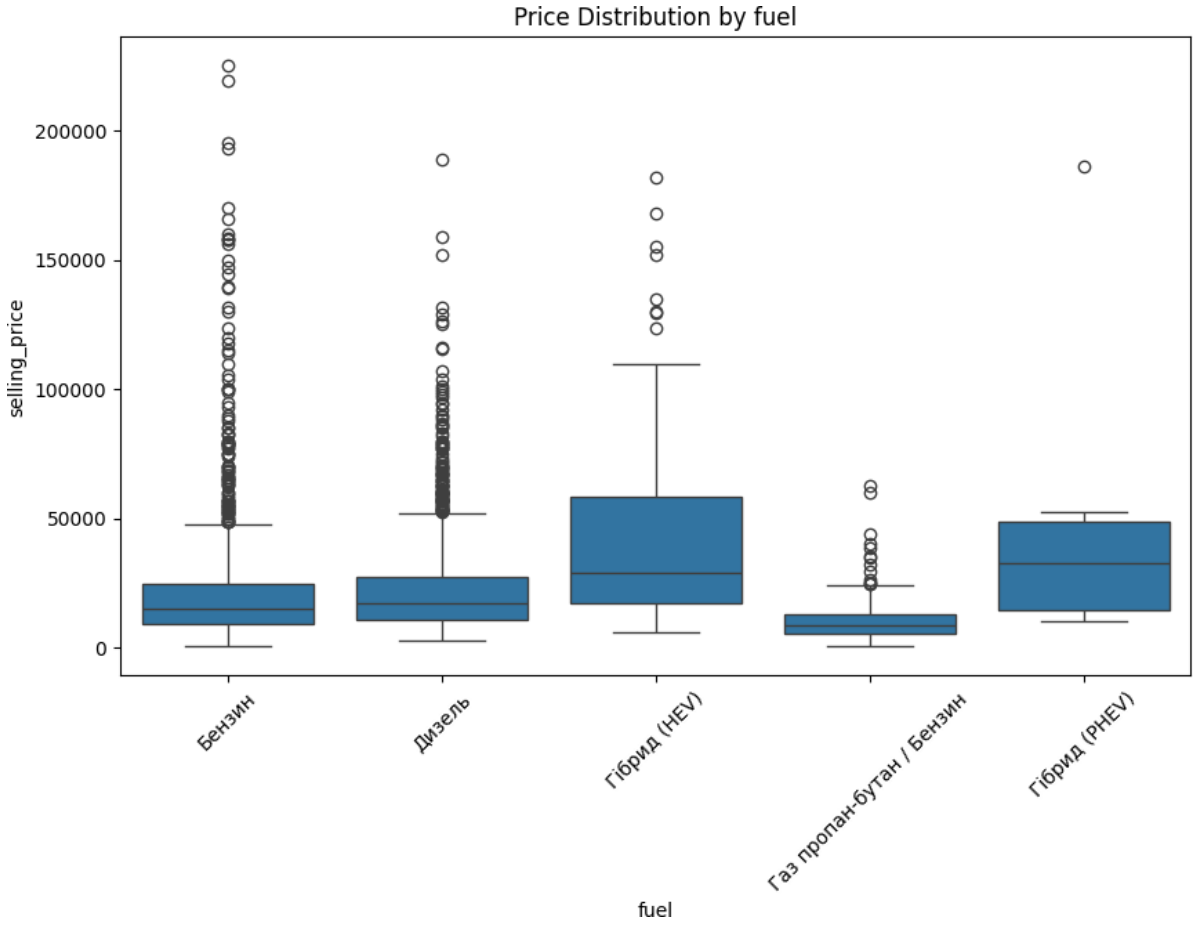

Рис. 3.5. Ціновий розподіл автомобілів за типом палива

Джерело: побудовано автором у Python

Ціна за кілометр пробігу в залежності від віку авто представлена рис. 3.5, де на горизонтальній осі знаходяться категорії за віком авто, а на вертикальній — ціни транспорту в доларах США. Даний графік показує середні 50 % цін у певній категорії. У середині кожного графіка є лінія, яка показує середнє значення ціни для кожної з категорій. Кружечками візуалізовано викиди.

З графіка на рис. 3.6 видно, що найдорожчими автомобілями є ті, вік яких не перевищує 3 років. З віком ціна автомобіля зменшується. Найбільше викидів знаходиться у категорії автомобілів віком 3-10 років.

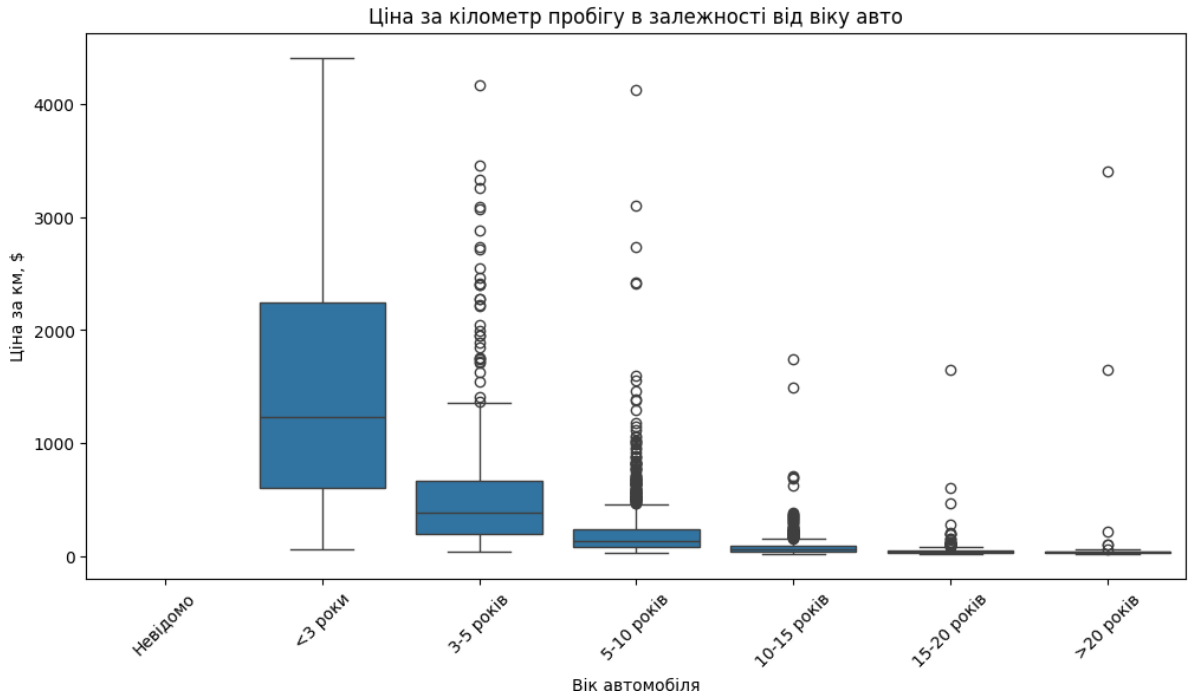

Рис. 3.6. Ціна за кілометр пробігу в залежності від віку авто

Джерело: побудовано автором у Python

Наступний графік (рис. 3.7) показує залежність ціни від пробігу у логарифмічній шкалі. По осі X зображено логарифм пробігу, а по осі Y — логарифм ціни.

Так як пробіг має негативний вплив на ціну, то зі збільшенням пробігу ціна автомобіля зменшується.

Найбільша щільність точок даних знаходиться в середньому діапазоні пробігів. Крім того, є викиди, де ціна не відповідає пробігу.

Цей графік показує, що зменшення ціни автомобіля відбувається логарифмічно, а не лінійно.

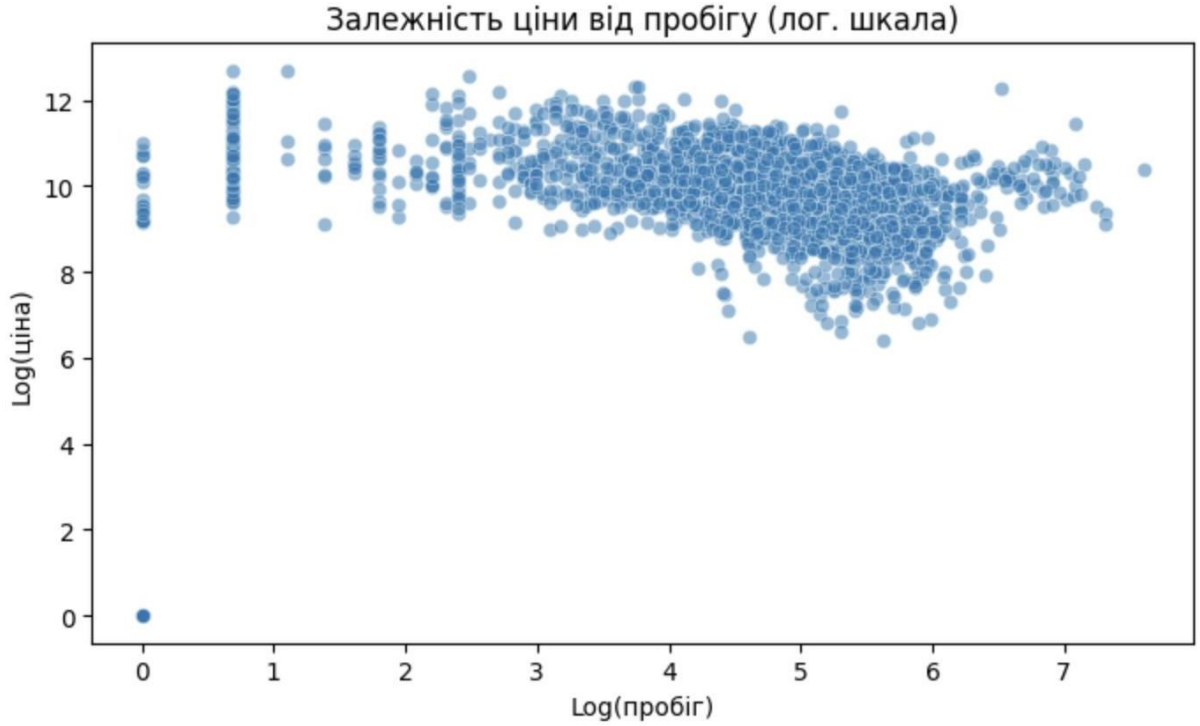

Рис. 3.7. Залежність ціни від пробігу у логарифмічній шкалі

Джерело: побудовано автором у Python

Далі проаналізуємо кореляційну матрицю (рис. 3.8), де позитивна кореляція показана теплим кольором, а негативна — холодним.

Найбільший вплив на ціну автомобіля чинить рік випуску автомобіля з кореляцією 0,53. Тобто чим новіший автомобіль, тим дорожча ціна. Негативна кореляція з віком автомобіля, яка є протилежною до кореляції з роком випуску — (-0,53). Помірна кореляція спостерігається з об’ємом двигуна — 0,32 — більший об’єм — вища ціна. Рік випуску ідеально протилежно корелює з віком автомобіля (-1), що є абсолютно логічним. Має негативну кореляцію з пробігом — (-0,45), що вказує на те, що старіші автомобілі мають більший пробіг. Спостерігається позитивна кореляція з пробігом — 0,46, тобто чим більший об’єм двигуна, тим більший пробіг. Таким чином, ціна найбільшим чином залежить від року автомобіля та об’єму двигуна.

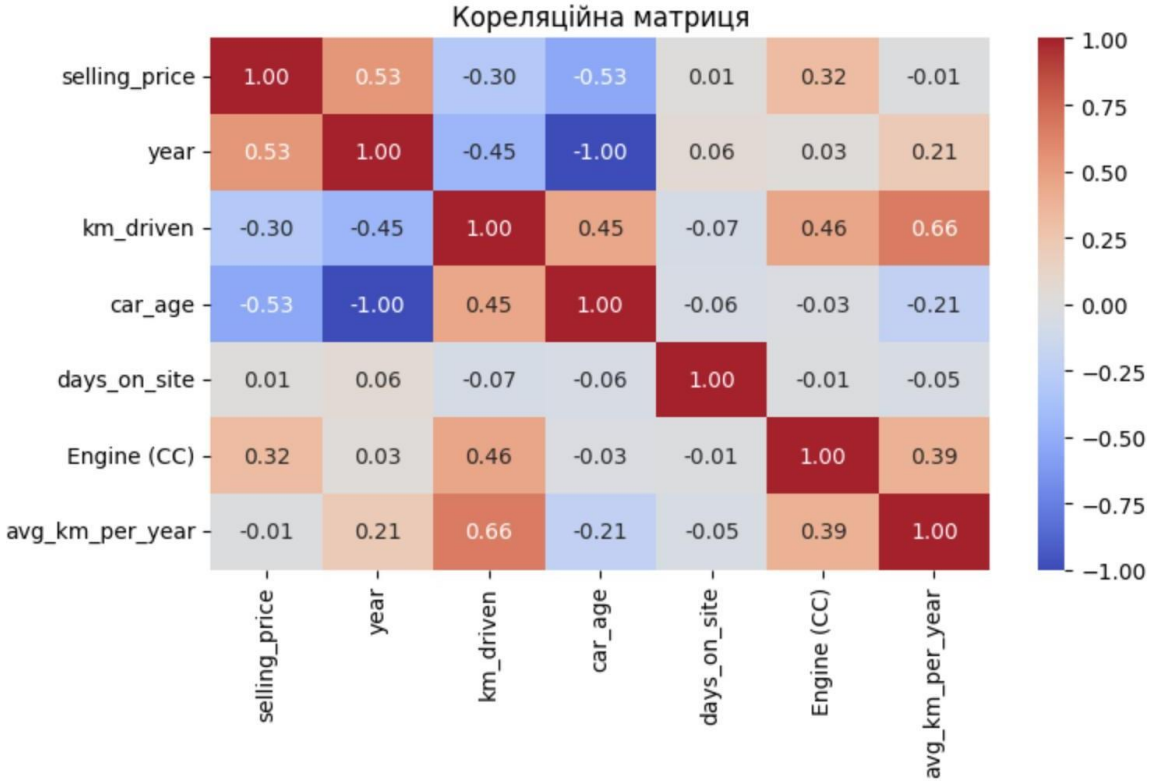

Рис. 3.8. Кореляційна матриця

Джерело: побудовано автором у Python

3.3. Детальна обробка даних

Спочатку здійснимо аналіз важливості ознак за допомогою такої функції:

```
feature_importance = pd.DataFrame({
    'feature': X.columns,
    'importance': model.feature_importances_
}).sort_values(by='importance', ascending=False)
```

	feature	importance
0	year	3.918615e-01
2	Engine (CC)	3.308396e-01
1	km_driven	1.866874e-01
267	fuel_Дизель	9.254521e-03
251	publication_date_2025-04-08	5.110959e-03
..	...	...
177	city_Сокаль	3.525591e-08
105	city_Левків	2.690072e-08
79	city_Кам'янка-Бузька	1.907018e-08
45	city_Вороновиця	1.890075e-08
26	city_Борщів	1.453551e-08

Рис. 3.9. Важливість ознак

Джерело: побудовано автором у Python

Найважливішою ознакою є рік з показником 0,392, що означає 39% від загальної важливості усіх ознак. Це логічно, так як на ціну автомобіля найбільше впливає саме його вік, спираючись на попередні результати.

Наступною важливою ознакою є об’єм двигуна, який займає 33% від усієї важливості. Це важлива технічна характеристика, на яку звертають увагу при купівлі транспорту.

Пробіг має важливість 19%, тобто він також є ключовим фактором, який впливає на вартість вживаного авто.

Наступні кроки полягали у:

1. **Обробці викидів:**

```
for col in columns:
  if method == 'iqr':
    Q1 = df_clean[col].quantile(0.25)
    Q3 = df_clean[col].quantile(0.75)
    IQR = Q3 — Q1
    lower_bound = Q1 — 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
```

```
                df_clean[col] = df_clean[col].clip(lower_bound, upper_bound)
```

2. Обробці пропущених значень:

```
    # Числові змінні
            numeric_cols = df_imputed.select_dtypes(include=['int64',
'float64']).columns
            if numeric_strategy == 'knn':
                df_imputed[numeric_cols] =
self.knn_imputer.fit_transform(df_imputed[numeric_cols])
            else:
                imputer = SimpleImputer(strategy=numeric_strategy)
                df_imputed[numeric_cols] =
imputer.fit_transform(df_imputed[numeric_cols])
    # Категоріальні змінні
            categorical_cols = df_imputed.select_dtypes(include=['object']).columns
            if len(categorical_cols) > 0:
                for col in categorical_cols:
                    imputer = SimpleImputer(strategy=categorical_strategy)
```

3. Нормалізації ознак:

```
            df_normalized = df.copy()
            numeric_cols = df_normalized.select_dtypes(include=['int64',
'float64']).columns
            if method == 'standard':
                df_normalized[numeric_cols] =
self.numeric_scaler.fit_transform(df_normalized[numeric_cols])
            elif method == 'robust':
                robust_scaler = RobustScaler()
                df_normalized[numeric_cols] =
robust_scaler.fit_transform(df_normalized[numeric_cols])
            elif method == 'power':
```

```
df_normalized[numeric_cols] =
self.power_transformer.fit_transform(df_normalized[numeric_cols])
```

Для базового прогнозу подивимося на розподіл реальних та прогнозованих значень за допомогою model.predict(X) — методу, який викликає функцію прогнозування моделі машинного навчання на наборі даних X.

Було отримано такі значення метрик для оцінки:

Mean Squared Error: 15573405.61 — це середній квадрат різниць між прогнозованими та фактичними значеннями. Ця метрика надає більшої ваги великим помилкам через квадратичну природу обчислень. Значення 15573405.61 саме по собі здається великим, але його треба інтерпретувати в контексті масштабу даних — цільова змінна має великі значення, то й MSE відповідно велике.

Root Mean Squared Error: 3946.32 — корінь квадратний з MSE, що має ту саму розмірність, що й цільова змінна. Це дозволяє легше інтерпретувати результати. Значення 3946.32 означає, що в середньому прогнози моделі відхиляються від фактичних значень приблизно на 3946 одиниць.

R-squared Score: 0.9727 — коефіцієнт детермінації, який показує, яку частку варіації залежної змінної пояснює модель. Значення 0.9727 означає, що модель пояснює 97.27% варіації даних, що є дуже хорошим показником. Чим ближче R-squared до 1, тим краще модель відповідає даним. Таке високе значення свідчить про відмінну прогностичну здатність моделі.

Mean Absolute Error: 1876.55 — це середнє абсолютних відхилень прогнозованих значень від фактичних. На відміну від MSE, ця метрика надає однакової ваги всім помилкам, незалежно від їх величини, тому вона менш чутлива до викидів. Значення 1876.55 означає, що в середньому прогнози відхиляються від реальних значень на 1876.55 одиниць, що не так багато, враховуючи, що ціни на вживані автомобілі встановлюються суб’єктивно, на основі того, як власник оцінює своє авто.

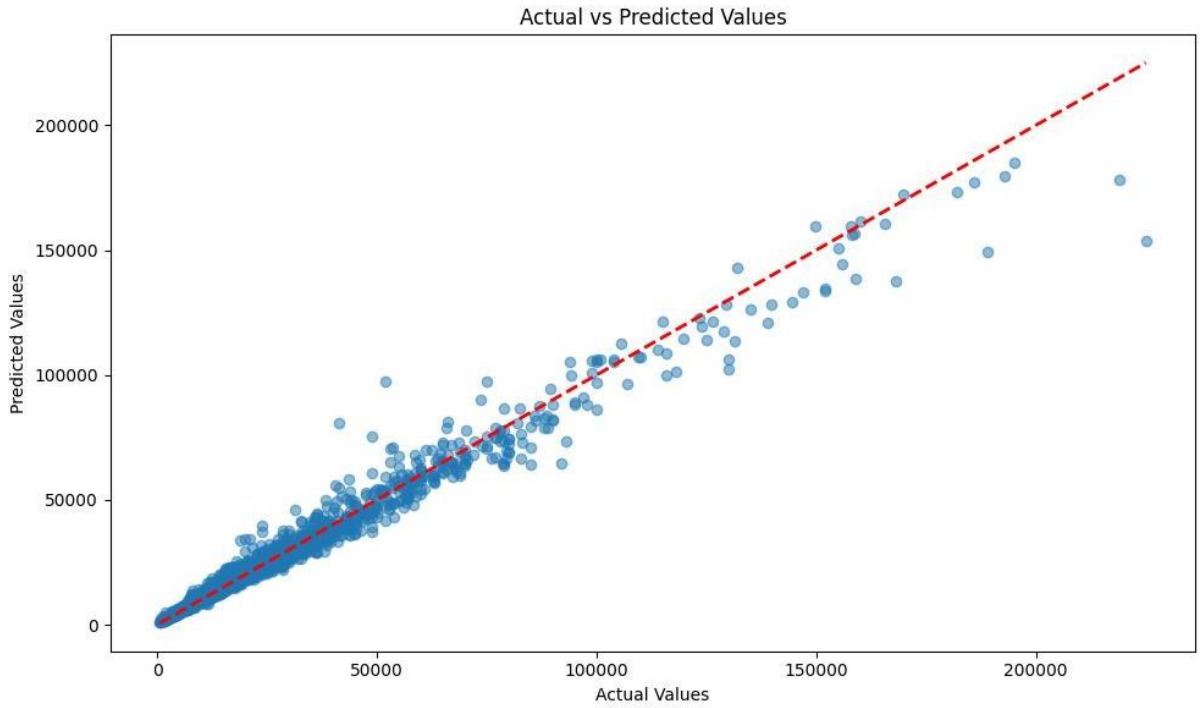

Рис. 3.10. Реальні та прогнозовані значення

Джерело: побудовано автором у Python

3.4. Порівняння базових моделей та вибір найкращої

Для наступного кроку додатково імпортуємо з sklearn модуль sklearn.ensemble, який містить ансамблеві методи машинного навчання, такі як:

- RandomForestClassifier
- RandomForestRegressor
- GradientBoostingClassifier
- BaggingClassifier

Модуль спеціалізується на методах, які об'єднують кілька моделей (дерев рішень, як правило), щоб покращити точність прогнозування.

```
----- НАЛАШТУВАННЯ ГІПЕРПАРАМЕТРІВ -----

Найкращі параметри для ridge:
Параметри: {'alpha': 10.0}
RMSE на крос-валідації: 0.6372

Найкращі параметри для lasso:
Параметри: {'alpha': 0.1}
RMSE на крос-валідації: 0.6806

Найкращі параметри для rf:
Параметри: {'max_depth': None, 'min_samples_split': 2, 'n_estimators': 200}
RMSE на крос-валідації: 0.1779

Найкращі параметри для gbm:
Параметри: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
RMSE на крос-валідації: 0.1486

Найкращі параметри для xgb:
Параметри: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
RMSE на крос-валідації: 0.1439

----- АНАЛІЗ НАЙКРАЩОЇ МОДЕЛІ -----
Найкраща модель за R²: xgb
Використовуємо налаштовану версію xgb

Статистика помилок:
mean_error: -0.0000
std_error: 0.1275
median_error: -0.0038
max_error: 0.8897
min_error: -1.6846
```

Рис. 3.11. Налаштування гіперпараметрів

Джерело: побудовано автором у Python

1. **XGBoost** показав найкращі результати з найнижчим RMSE на тестових даних (0.1235) та найвищим R^2 (0.9846) — модель пояснює 98.46% варіативності цін. Також модель має низький MAE на рівні 0.0598.
2. **Random Forest** показав дуже близькі результати з RMSE = 0.1411, R^2 = 0.9799, а також з найнижчий середнім абсолютним відхилень прогнозованих значень (MAE) — 0.0546
3. **Gradient Boosting** теж продемонстрував хороші результати, де корінь квадратний з MSE (RMSE) дорівнює 0.2455, коефіцієнт детермінації — 0.9390 та MAE — 0.1491.
4. **Лінійні моделі** показали значно гірші результати:
    - Для Ridge R^2 = 0.5680

- Lasso практично не навчився (R^2 близько 0)
- Linear Regression показала катастрофічно погані результати, що вказує на проблеми з даними або мультиколінеарність, тобто лінійна регресія не здатна вирішити задачу і сформувати адекватні прогнози.

Підбір гіперпараметрів покращив усі моделі:

1. **XGBoost** — найкращі параметри:
   - 200 дерев
   - learning_rate 0.1
   - max_depth 5
   - Покращення RMSE до 0.1439 на крос-валідації
2. **Gradient Boosting**:
   - Аналогічні параметри як у XGBoost
   - RMSE 0.1486 на крос-валідації
3. **Random Forest**:
   - 200 дерев
   - Необмежена глибина
   - RMSE 0.1779 на крос-валідації

3.5. Навчання моделі та аналіз результатів

Для навчання використовувала модель RandomForestRegressor, яка показала одні з найкращих результатів на тестових даних.

Перенавчання моделі на тренувальних даних:

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
def predict_car_price(input_data, model, X_columns):
    input_df = pd.DataFrame([input_data])
    input_df = pd.get_dummies(input_df)
    for col in X_columns:
        if col not in input_df.columns:
```

```
            input_df[col] = 0
    input_df = input_df[X_columns]
    predicted_price = model.predict(input_df)[0]
    return predicted_price
```

Створюю тестовий приклад автомобіля:

```
test_car = {
        'brand': 'Audi A6',
        'year': 2017,
        'km_driven': 45000,
        'city': 'Київ',
        'Engine (CC)': 3.00,
        'fuel': 'Бензин',
}
```

Прогнозована ціна автомобіля: 28,145.68

Метрики якості на тестовій вибірці:

- **RMSE**: 11,480.25
- **R^2**: 0.7958

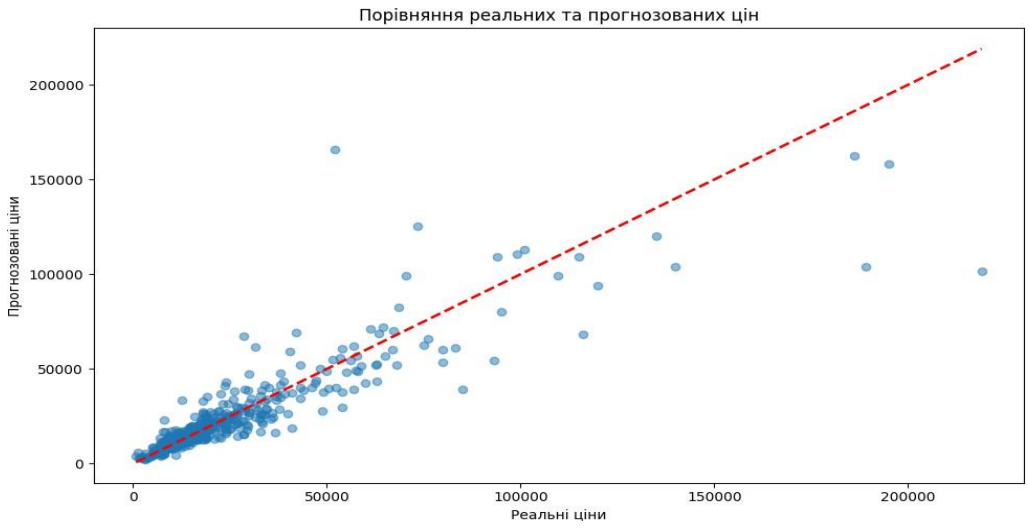

Рис. 3.12. Порівняння реальних та прогнозованих значень

Джерело: побудовано автором у Python

На рис. 3.12 видно, що більшість точок знаходиться біля червоної лінії, що вказує на якість тренувальної моделі. Звісно є викиди, які знаходяться біля

цінового діапазону вище 100 000$, що пояснюється малою вибіркою дорогих авто, так як здебільшого на таких платформах, як auto.ria розміщують оголошення про автомобілі низького та середнього класів.

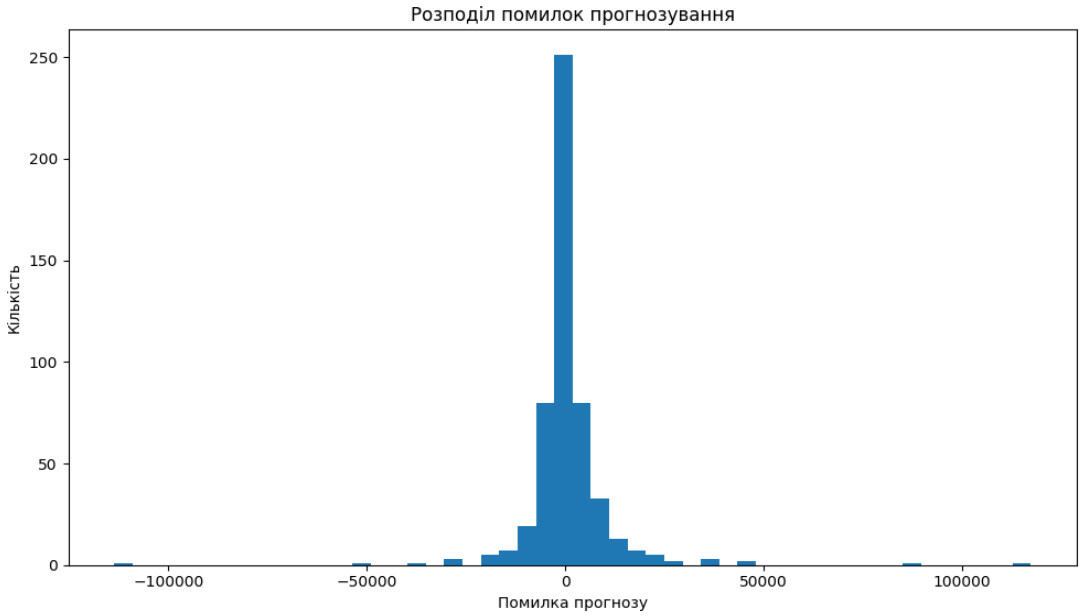

Рис. 3.13. Розподіл помилок прогнозування

Джерело: побудовано автором у Python

На рис. 3.13 зображено розподіл помилок прогнозування моделі, що представлений у вигляді гістограми.

Вісь X представляє помилку прогнозу — різницю між прогнозованими та фактичними значеннями. Від'ємні значення означають, що модель дає прогноз менший за реальне значення, а додатні — що модель переоцінює значення.

Вісь Y показує кількість спостережень, які потрапляють у відповідний діапазон помилок.

Графік має виражену дзвоноподібну форму з центром близько до нуля, що свідчить про нормальний розподіл помилок — більшість помилок сконцентрована навколо нуля. Найвища частота спостерігається при помилках близьких до нуля (приблизно 250 спостережень), що вказує на те, що модель часто дає досить точні прогнози.

Розподіл є майже симетричним, але з невеликим зміщенням — більше спостережень мають невеликі від'ємні помилки, ніж додатні. Це може свідчити про те, що модель має незначну тенденцію до недооцінки значень.

Діапазон помилок прогнозування достатньо широкий — від приблизно -100000 до +100000, але більшість помилок знаходиться в межах ±20000, що узгоджується з раніше розрахованим RMSE у 3946.32

Аналіз відносних помилок показав наступні результати:

- **Середня відносна помилка**: 22.73%.
- **Медіанна відносна помилка**: 15.08%.

Середня відносна помилка складає 22.73%. Це означає, що в середньому модель відхиляється від реальних значень приблизно на 23% від фактичного значення. Такий показник дає краще уявлення про точність моделі відносно масштабу прогнозованих значень, ніж абсолютні метрики (як RMSE або MAE).

Медіанна відносна помилка становить 15.08%. Медіана нижча за середнє значення, що свідчить про асиметричний розподіл відносних помилок — більшість прогнозів має нижчі відносні помилки, ніж середня. Ця різниця між середнім і медіаною (приблизно 7.65 процентних пунктів) вказує на наявність викидів з досить високими відносними помилками, які "тягнуть" середнє значення вгору. Це знову ж таки пояснюється наявністю одиничних оголошень з дуже високими цінами.

Порівнюючи ці відносні помилки з високим R^2 (0.9727), можна зробити висновок, що модель загалом дає хороші прогнози на рівні тенденцій і закономірностей, але для окремих випадків відхилення можуть бути суттєвими, тому модель потребує майбутнього доопрацювання з додаванням більшої кількості факторів.

Результат

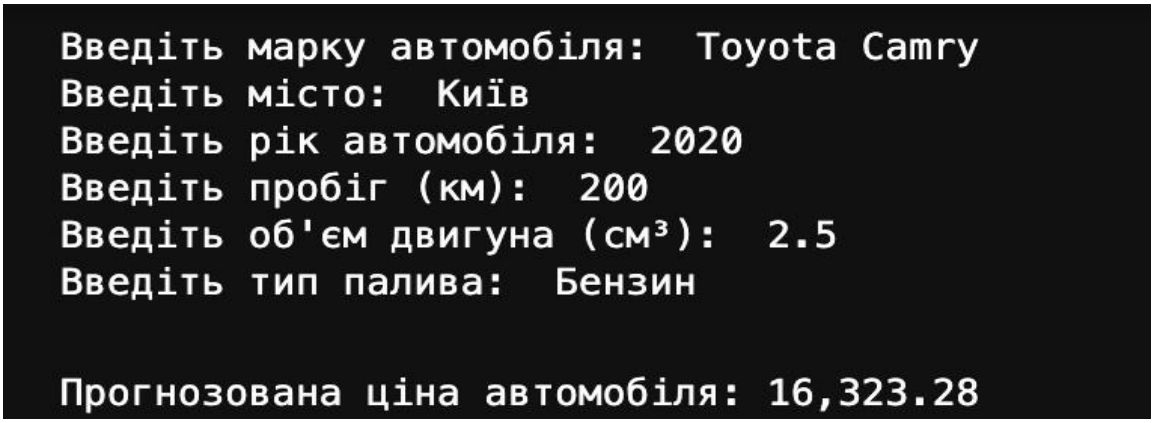

Рис. 3.14. Результат моделі

Джерело: побудовано автором у Python

На рис. 3.14 зображено ввід та вивід моделі, якщо ж знайти в Інтернеті вживаний автомобіль з такими параметрами, то ціна буде орієнтовно такою ж. Це свідчить про те, що модель чудово впоралася з поставленим завданням. Подальші дослідження будуть спрямовані на вдосконалення моделі.

ВИСНОВОК

У ході виконання бакалаврської роботи було успішно реалізовано повний цикл побудови моделі, яка може прогнозувати ціну вживаного автомобіля. Виконано усі поставлені завдання: від аналізу ринку до створення автоматизованої системи. Виявилося, що Україна має один з найнижчих Індексів прозорості на автомобільному ринку серед європейських країн. Це вказує на те, що для споживачів існує ризик купити авто за завищеною ціною, тому необхідно створювати нові інструменти контролю та оцінки вартості.

Модель створювалася за допомогою інструментів машинного навчання і використовувала реальні дані з української платформи auto.ria. Вона здатна зібрати найновіші дані з сайту та швидко їх обробити.

Таким чином, було здійснено збір, очистку даних від викидів та їх обробку за допомогою бібліотек Python. Розвідувальний аналіз дозволив виявити важливі зв’язки між ціною та характеристиками автомобільного транспорту.

Запропонована модель може працювати в режимі реального часу та має високу пояснювальність близько 97 %. Вона дозволить користувачам отримати реальну оцінку транспортного засобу, зменшить інформаційну асиметрію та допоможе у виявленні потенційно шахрайських оголошень.

Таким чином її можна застосовувати як звичайним людям, які хочуть придбати/продати автомобіль, так і страховим компаніям, банкам або торговим платформам. Така модель відкриває перспективи для подальших досліджень і вдосконалень алгоритмів, зокрема для врахування інших факторів, наприклад, ДТП чи кількості власників.

Виконане дослідження підтвердило ефективність застосування прогнозної аналітики та машинного навчання у галузі продажів автомобілів на вторинному ринку. Це може стати важливим кроком для підвищення прозорості, зменшення ризику натрапити на шахраїв та сприятиме рівню довіри користувачів до автомобільного ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. An Analysis of Car Price Prediction using Machine Learning. ACM Digital Library. URL: https://dl.acm.org/doi/10.1145/3674029.3674032 (дата звернення: 01.02.2025).

2. Predicting Car Prices Using Machine Learning and Data Science. Medium. URL: https://medium.com/odscjournal/predicting-car-prices-using-machine-learning-and-data-science-52ed44abab1b (дата звернення: 01.02.2025).

3. Car Price Prediction. ACM Digital Library. URL: https://dl.acm.org/doi/10.1145/3590837.3590864 \ (дата звернення: 01.02.2025).

4. Predicting the Price of Used Cars using Machine Learning Techniques. ResearchGate. URL: https://www.researchgate.net/publication/319306871_Predicting_the_Price_of_Used_Cars_using_Machine_Learning_Techniques (дата звернення: 01.02.2025).

5. Шевченківська весна 2025. Економічний факультет Київського національного університету імені Тараса Шевченка.

6. Auto.Ria. URL: https://auto.ria.com/uk/.

7. Найбільший автовиробник – Китай: скільки автівок виготовили в світі торік. Економічна правда. URL: https://epravda.com.ua/biznes/skilki-avtotransportu-virobili-v-sviti-u-2024-roci-804582/ (дата звернення: 01.02.2025).

8. Про мобілізаційну підготовку та мобілізацію. Верховна Рада України. URL: https://zakon.rada.gov.ua/laws/show/3543-12#n662 (дата звернення: 01.02.2025).

9. Український авторинок — прогноз на 2025 рік від Інституту досліджень авторинку. Інститут досліджень авторинку. URL: https://eauto.org.ua/news/698-ukrajinskiy-avtorinok-prognoz-na-2025-rik-vid-institutu-doslidzhen-avtorinku (дата звернення: 01.03.2025)

10. Змінюємо підхід до купівлі-продажу авто: у Дії можна перереєструвати авто онлайн. Дія. URL: https://diia.gov.ua/news/zminyuyemo-pidhid-do-kupivli-

prodazhu-avto-u-diyi-mozhna-perereyestruvati-avto-onlajn (дата звернення: 01.03.2025).

11. Used Vehicle Value Index. Manheim. URL: https://site.manheim.com/en/services/consulting/used-vehicle-value-index.html (дата звернення: 01.03.2025).

12. Трекер економіки України під час війни. Центр економічної стратегії. URL: https://ces.org.ua/tracker-economy-during-the-war/?gad_source=1&gclid=CjwKCAjwzMi_BhACEiwAX4YZUG2D2gas6a-4fBdjx9ctnByKuSy8p6E06l7mMeVqdNX4VsCqmmAqkRoCM-EQAvD_BwE (дата звернення: 01.03.2025).

13. Де дешевше, а де дорожче? Цінова картина автовторинку на початку року. Інститут досліджень авторинку. URL: https://eauto.org.ua/news/730-de-deshevshe-a-de-dorozhche-cinova-kartina-avtovtorinku-na-pochatku-roku (дата звернення: 01.03.2025).

14. Індекс прозорості ринку вживаних автомобілів України. CarVertical. URL: https://www.carvertical.com/ua/transparency-index/ukraine (дата звернення: 01.03.2025).

15. Predictive Analytics: Definition, Model Types, and Uses. Investopedia. URL: https://www.investopedia.com/terms/p/predictive-analytics.asp.

16. Decision Trees. Scikit-learn. URL: https://scikit-learn.org/stable/modules/tree.html.

17. 7 Regression Techniques You Should Know! Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/.

18. What is a neural network? IBM. URL: https://www.ibm.com/think/topics/neural-networks.

19. Класичне машинне навчання: завдання класифікації, узагальнення, кластеризації даних. Evergreens. URL: https://evergreens.com.ua/ua/articles/classical-machine-learning.html.

20. Predictive Modeling & Statistical Analysis: Leveraging Data Science to Make Better Decisions. New York Institute of Technology. URL: https://online.nyit.edu/blog/predictive-modeling-statistical-analysis-leveraging-data-science-to-make-better-decisions.

21. What is exploratory data analysis (EDA)? IBM. URL: https://www.ibm.com/think/topics/exploratory-data-analysis.

22. Random Forest Classifier and its Hyperparameters. Medium. URL: https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6.

23. Gradient Boosting in ML. Geeksforgeeks. URL: https://www.geeksforgeeks.org/ml-gradient-boosting/.

24. Python: Requests Library [Beginner]. Medium. URL: https://olivierkonate.medium.com/python-requests-library-beginner-60f59112c71d.

25. Beautiful Soup Documentation. Crummy. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

26. Pandas (software). WikipediA. URL: https://en.wikipedia.org/wiki/Pandas_(software).

27. Pandas Introduction. W3schools. URL: https://www.w3schools.com/python/pandas/pandas_intro.asp.

28. NumPy Introduction. W3schools. URL: https://www.w3schools.com/python/numpy/numpy_intro.asp.

29. SciPy. WikipediA. URL: https://en.wikipedia.org/wiki/SciPy.

30. Statsmodels. Codecademy. URL: https://www.codecademy.com/resources/docs/python/statsmodels.

31. What is Scikit-Learn (Sklearn)? IBM. URL: https://www.ibm.com/think/topics/scikit-learn.

32. Introduction to Matplotlib. Geeksforgeeks. URL: https://www.geeksforgeeks.org/python-introduction-matplotlib/.

33. Introduction to Seaborn - Python. Geeksforgeeks. URL: https://www.geeksforgeeks.org/introduction-to-seaborn-python/.

34. Цінові особливості вторинного ринку легкових автомобілів Україні: регіональний аналіз. West Auto Hub. URL: https://wah.ua/blog/183-cinovi-osoblivosti-vtorinnogo-rinku-legkovix-avtomobiliv-v-ukrayini-regionalnii-analiz.

35. Стандартна бібліотека Python. Python. URL: https://docs.python.org/uk/3.13/library/index.html.