БАКАЛАВРСЬКА РОБОТА

КРБ.ІСТ-17.00.00.000 ПЗ

Група ІСТ-22-1

Микола СЛЮСАРЧУК

2026

Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інформаційно-телекомунікаційних технологій та систем

Слюсарчук Микола Тарасович

(прізвище, ім'я, по батькові)

УДК 004.85:519.2

(індекс)

БАКАЛАВРСЬКА РОБОТА

Розроблення інформаційної системи прогнозування ціни автомобілів на основі їх характеристик з використанням методів машинного навчання

(назва роботи)

Інформаційні системи та технології

(назва освітньої програми)

126 − Інформаційні системи та технології

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня Слюсарчук М. Т.

(підпис, ініціали та прізвище здобувача)

Науковий керівник *Паньків Х. В., к.т.н., доц. каф.ІТТС*

(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
В. о. завідувача кафедри

Заміховська О. Л.

(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2026

Івано-Франківський національний технічний університет нафти і газу
(повне найменування закладу вищої освіти)

Факультет інформаційних технологій
Кафедра інформаційно-телекомунікаційних технологій та систем
Освітній рівень бакалавр
Спеціальність 126 − Інформаційні системи та технології
(шифр і назва)

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ІТТС
Заміховська О. Л.
«____» _____2026 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Слюсарчуку Миколі Тарасовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Розроблення інформаційної системи прогнозування ціни автомобілів на основі їх характеристик з використанням методів машинного навчання*
керівник роботи, *Паньків Х. В., к.т.н., доц. каф.ІТТС*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від " 07 " *квітня* 2026 року № 163/7
2. Строк подання студентом роботи
3. Вихідні дані до роботи Інформаційна система прогнозування ринкової вартості автомобілів методами машинного навчання; мова Python; відкритий набір даних Kaggle (10 000 записів, 10 ознак); передобробка ознак (One-Hot Encoding, RobustScaler); побудова й порівняння одинадцяти регресійних моделей (scikit-learn, XGBoost, LightGBM, CatBoost); оптимізація гіперпараметрів (GridSearchCV); оцінювання за метриками MAE, MSE, RMSE, R^2; вебзастосунок на фреймворку Streamlit.
4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)
дослідження предметної області, огляд аналогів та постановка задачі; огляд методів і метрик машинного навчання та проєктування архітектури системи; програмна реалізація: аналіз даних, навчання й порівняння моделей, розробка вебзастосунку.
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
схема пайплайну передобробки даних та навчання моделей; порівняльна таблиця точності регресійних моделей за метриками MAE, MSE, RMSE та R^2; архітектурна схема вебзастосунку AutoPrice AI та інтерфейси користувача.
6. Дата видачі завдання: 07.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Термін виконання етапів роботи	Примітка
1	Дослідження предметної області та постановка завдання	Березень, 2026	виконано
2	Розробка структури інформаційної системи	Квітень, 2026	виконано
3	Розробка програмного забезпечення	Травень, 2026	виконано
4	Оформлення пояснювальної записки	Червень, 2026	виконано

Студент _____ *Слюсарчук М. Т.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ Паньків Х. В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

У рамках бакалаврської кваліфікаційної роботи розроблено інформаційну систему прогнозування ринкової вартості автомобілів на основі їхніх технічних та експлуатаційних характеристик із використанням методів машинного навчання.

У результаті створено повнофункціональну систему, що складається з предиктивного ядра для обробки даних і навчання моделей та інтерактивного вебзастосунку AutoPrice AI для взаємодії з кінцевим користувачем. Систему навчено на відкритому наборі даних обсягом 10 000 записів, що містить десять ознак автомобіля. Проведено розвідувальний аналіз даних (EDA), конструювання й кодування ознак методом One-Hot Encoding та масштабування числових характеристик за допомогою RobustScaler. Побудовано та порівняно одинадцять регресійних моделей — від лінійної регресії й методу k-найближчих сусідів до ансамблевих алгоритмів градієнтного бустингу (XGBoost, LightGBM, CatBoost) — за метриками MAE, MSE, RMSE та R^2.

Серверну частину реалізовано мовою Python із застосуванням бібліотек pandas, NumPy та scikit-learn; найкращі за точністю результати продемонстрували ансамблеві моделі, а для розгортання у застосунку обрано лінійну модель Lasso як оптимальний компроміс між точністю та інтерпретованістю. Інтерактивний вебінтерфейс побудовано на фреймворку Streamlit, що дозволило користувачеві вводити характеристики автомобіля та миттєво отримувати прогноз його ринкової вартості. Розроблена система усуває вплив суб'єктивного людського фактора та забезпечує швидку й обґрунтовану оцінку вартості транспортного засобу.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ПРОГНОЗУВАННЯ ЦІНИ, МАШИННЕ НАВЧАННЯ, РЕГРЕСІЯ, АВТОМОБІЛЬНИЙ РИНОК, ВЕБЗАСТОСУНОК, PYTHON, SCIKIT-LEARN, CATBOOST, STREAMLIT.

ABSTRACT

Within the framework of the bachelor's qualifying work, an information system for predicting the market value of cars based on their technical and operational characteristics was developed using machine learning methods.

As a result, a fully functional system was created, consisting of a predictive core for data processing and model training and an interactive web application, AutoPrice AI, for end-user interaction. The system was trained on an open dataset of 10,000 records containing ten car features. Exploratory data analysis (EDA), feature engineering and One-Hot Encoding, and the scaling of numerical features using RobustScaler were performed. Eleven regression models were built and compared — from linear regression and the k-nearest neighbors method to ensemble gradient boosting algorithms (XGBoost, LightGBM, CatBoost) — using the MAE, MSE, RMSE, and R^2 metrics.

The back-end part was implemented in Python using the pandas, NumPy, and scikit-learn libraries; the ensemble models demonstrated the best accuracy, while the linear Lasso model was chosen for deployment as an optimal compromise between accuracy and interpretability. The interactive web interface was built with the Streamlit framework, allowing the user to enter car characteristics and instantly obtain a prediction of its market value. The developed system eliminates the influence of subjective human factors and provides a fast and well-grounded estimation of a vehicle's value.

KEYWORDS: INFORMATION SYSTEM, PRICE PREDICTION, MACHINE LEARNING, REGRESSION, AUTOMOTIVE MARKET, WEB APPLICATION, PYTHON, SCIKIT-LEARN, CATBOOST, STREAMLIT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних

ІС – інформаційна система

ML – Machine Learning (машинне навчання)

AI – Artificial Intelligence (штучний інтелект)

EDA – Exploratory Data Analysis (розвідувальний аналіз даних)

LR – Linear Regression (лінійна регресія)

KNN – k– Nearest Neighbors (метод k найближчих сусідів)

CART – Classification and Regression Tree (дерево рішень)

RF – Random Forest (випадковий ліс)

GBM – Gradient Boosting Machine (градієнтний бустинг)

MAE – Mean Absolute Error (середня абсолютна похибка)

MSE – Mean Squared Error (середня квадратична похибка)

RMSE – Root Mean Squared Error (корінь із середньої квадратичної похибки)

R^2 – Coefficient of Determination (коефіцієнт детермінації)

CPU – Central Processing Unit (центральний процесор)

GPU – Graphics Processing Unit (графічний процесор)

CSV – Comma– Separated Values (формат даних із роздільниками–комами)

USD – United States Dollar (долар США)

ВСТУП

Актуальність. Автомобільний ринок залишається одним із найбільш динамічних сегментів товарного ринку, а питання об’єктивного визначення вартості транспортних засобів має критичне значення для приватних покупців і продавців, автодилерів, страхових та фінансових установ. Ціна автомобіля формується під впливом великої кількості взаємопов’язаних чинників: марки й моделі, року випуску, пробігу, об’єму двигуна, типу палива та коробки передач, кількості попередніх власників, і постійно змінюється разом із ринковою кон’юнктурою. Масштаби національного автопарку та висока активність вторинного ринку унеможливлюють якісну ручну оцінку кожної пропозиції, а традиційні експертні методи, що спираються на суб’єктивний досвід чи порівняння з кількома оголошеннями, призводять до значних похибок і фінансових ризиків. Це зумовлює потребу в автоматизованих інтелектуальних інструментах, здатних на основі великих масивів даних надавати точний і прозорий прогноз вартості. Саме тому розроблення інформаційної системи прогнозування ринкової вартості автомобілів на основі методів машинного навчання є актуальним науково–практичним завданням.

Мета роботи – розроблення інформаційної системи прогнозування ринкової вартості автомобілів, яка на основі методів машинного навчання забезпечує точну та доступну оцінку ціни транспортного засобу за його технічними й експлуатаційними характеристиками, та її практична реалізація у вигляді інтерактивного вебзастосунку.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– дослідити предметну область, чинники ціноутворення на ринку автомобілів та існуючі програмні аналоги;

– підготувати набір даних: провести розвідувальний аналіз, очищення, конструювання та кодування ознак, масштабування числових характеристик;

– розглянути й порівняти основні методи машинного навчання для задачі регресії та обрати найточнішу модель за метриками MAE, MSE, RMSE та R^2;

– виконати налаштування гіперпараметрів обраних моделей, проаналізувати точність прогнозів та важливість ознак;

– розробити вебзастосунок, що надає користувачеві зручний інтерфейс для введення характеристик автомобіля та отримання прогнозованої ціни;

– оцінити практичне значення розробленої системи.

Об'єкт дослідження – процес визначення (прогнозування) ринкової вартості автомобіля на основі сукупності його характеристик.

Предмет дослідження – методи та програмні засоби машинного навчання для побудови регресійної моделі прогнозування ціни автомобіля та її розгортання у вигляді вебзастосунку.

Методи дослідження. У роботі використано методи аналізу та узагальнення літературних джерел, статистичні методи розвідувального й кореляційного аналізу даних, методи машинного навчання (регресійні та ансамблеві моделі) та об'єктно– орієнтованого програмування для реалізації застосунку.

Практичне значення. Розроблена система може бути використана учасниками автомобільного ринку – покупцями, продавцями, автодилерами, страховими й фінансовими установами – для швидкої та обґрунтованої оцінки вартості автомобіля, а також як основа для подальшого вдосконалення предиктивних моделей на ширших наборах даних.

Структура роботи обумовлена метою та поставленими завданнями. Бакалаврська кваліфікаційна робота складається зі вступу, трьох розділів з підрозділами та висновками до кожного, загальних висновків, списку використаних джерел і додатків. Робота містить таблиці та рисунки, що ілюструють результати дослідження.

1 ДОСЛІДЖЕНННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність процесу прогнозування ціни автомобілів

У сучасних умовах розвитку цифрової економіки автомобільний ринок залишається одним із найбільш динамічних сегментів товарного ринку. Питання об’єктивного визначення вартості транспортних засобів має критичне значення для широкого кола учасників: від приватних покупців та продавців до професійних автодилерів, страхових компаній та банківських установ. Складність задачі полягає у тому, що ціна автомобіля не є статичною величиною; вона формується під впливом великої кількості переплетених чинників, що постійно змінюються. Особливої ваги проблема точного оцінювання набуває через масштабність національного автопарку. Згідно з аналітичними даними, станом на початок 2024 року в Україні нараховується близько 10,5 мільйонів автомобілів, а рівень автомобілізації демонструє стабільне зростання, досягнувши показника у 250 машин на 1000 жителів [1].

Такий обсяг ринку унеможливлює якісну ручну оцінку кожної пропозиції, що створює гостру потребу в автоматизованих інтелектуальних інструментах. Важливість розроблення предиктивних моделей підкріплюється динамікою вторинного ринку. Попри економічні виклики, активність у сегменті вживаних авто залишається високою. За результатами аналізу ринкових трендів, середня ціна вживаного автомобіля на внутрішньому ринку України на початку 2025 року зафіксувалася на рівні близько 6000 доларів США. На рисунку 2.1 зображено зміни середньої ціни на ринку протягом року[2].

Рисунок 1.1 – Зміна середньої ціни пропозицій на ринку

Графік демонструє коливання вартості з помітним падінням у першій половині року та досягненням мінімуму в літньо–осінній період. Найбільш суттєве зростання зафіксовано наприкінці 2024 року, коли ціна сягнула позначки у 6 500 $, проте цей тренд змінився корекцією у бік зниження в січні 2025 року.

Проте ця цифра є усередненою, оскільки реальна вартість конкретного об'єкта залежить від його індивідуальних параметрів, а ринкові коливання вимагають постійної перевірки актуальності ціни.

Основними факторами, що впливають на формування ціни та мають бути враховані моделлю, є:

- Технічні характеристики: марка та модель, рік випуску (вік авто), загальний пробіг, об’єм та потужність двигуна.
- Експлуатаційні параметри: тип палива, вид трансмісії, кількість попередніх власників та історія обслуговування.
- Ринкові умови: поточний попит на конкретну модель, сезонність та загальна економічна ситуація в країні.

Традиційні методи оцінки, що базуються на суб’єктивному досвіді експертів або порівнянні з кількома аналогічними оголошеннями, часто призводять до значних похибок. Це зумовлює фінансові ризики: продавець може втратити кошти через занижену ціну, а покупець – переплатити за авто в незадовільному стані.

Отже, розроблення інформаційної системи, здатної на основі великих масивів даних мінімізувати вплив людського фактора та надавати точний прогноз вартості, є актуальним науково–практичним завданням. Використання сучасних методів аналізу дозволяє забезпечити прозорість ціноутворення та оптимізувати процеси прийняття рішень на насиченому та мінливому автомобільному ринку України.

Врахування взаємозв’язків між цими параметрами традиційними експертними методами часто є суб’єктивним, що знижує точність оцінки та призводить до фінансових втрат учасників ринку. У зв’язку з цим зростає інтерес до використання методів машинного навчання, які дозволяють автоматично виявляти приховані закономірності у великих обсягах даних. Задача прогнозування ціни у цьому контексті формулюється як задача регресії, де цільовою змінною є вартість, а вхідними параметрами – технічні характеристики об'єкта.

Наукові дослідження підтверджують ефективність застосування алгоритмів лінійної регресії, дерев рішень та випадкового лісу для мінімізації похибки прогнозування. Автоматизація цього процесу за допомогою інтелектуальних інформаційних систем сприяє зменшенню впливу людського фактора та підвищенню прозорості ринку.

Таким чином, розроблення інформаційної системи на основі машинного навчання є актуальним науково-практичним завданням. Враховуючи масштаби українського автопарку та стабільний попит на вторинному ринку зі сформованими ціновими трендами, створення інструменту для високоточної оцінки вартості відповідає сучасним потребам цифрової економіки та запитам бізнесу.

1.2 Основні поняття автомобільної галузі

Автомобільний ринок – це надзвичайно багатопланова сфера. Кожного дня транспортні засоби переходять від продавців до покупців за ціною, що встановлюється під впливом великої кількості чинників. Вартість машини не є статичною: вона формується на основі низки параметрів і може змінюватися з часом під дією тих самих факторів. Для кращого розуміння ціноутворення в автоіндустрії варто детально розглянути ключові терміни та характеристики.

Марка автомобіля – це найменування виробника, який створив транспортний засіб. Вона слугує головним ідентифікатором, оскільки втілює в собі корпоративну стратегію, загальну концепцію, дизайнерську мову, фірмовий стиль та унікальні інженерні здобутки конкретної компанії. Серед найбільш впізнаваних брендів можна назвати BMW, Toyota, Ford, Mercedes–Benz, Honda, Audi, Chevrolet та багато інших. Кожна марка зазвичай орієнтується на певні ринкові ніші або спеціалізується на конкретних типах авто: від спорткарів та позашляховиків до бюджетних міських моделей чи авто преміум сегмента. Репутація бренду, побудована на якості, довговічності, стилі, легкості в обслуговуванні та впровадженні інновацій, є одним із визначальних критеріїв під час вибору машини.

Модель автомобіля представляє собою конкретну версію або лінійку машин, що випускаються брендом. Моделі різняться між собою зовнішнім виглядом, технічними даними, габаритами, матеріалами оздоблення та конструктивними рішеннями. Кожна модель зазвичай має власну назву, яка підкреслює її призначення чи особливості. Наприклад, у модельному ряду BMW є такі різні представники, як BMW 3 Series, BMW X5 чи електрокар BMW i8. Крім того, одна модель може пропонуватися в різних типах кузова (як– от седан, хетчбек чи купе), з різними типами двигунів (електричні, дизельні, бензинові) та варіантами приводу (повний або передній). Вибір моделі зумовлений особистими потребами покупця, адже кожна з них орієнтована на свою цільову групу: молодь частіше обирає динамічні й швидкі

авто, тоді як старші люди віддають перевагу надійності та практичності. Розуміння відмінностей між моделями дозволяє обрати оптимальний варіант, що відповідає бюджету та способу життя.

Рік випуску – це дата виготовлення конкретної одиниці транспорту на заводі. Цей показник є критично важливим для визначення ціни. Оскільки світові автовиробники зазвичай оновлюють технології та дизайн щороку або через певні цикли, новіші машини є більш досконалими технічно, мають краще оснащення та сучасний вигляд. Важливим аспектом є також сервісна підтримка: чим старіше авто, тим складніше знайти для нього оригінальні запчастини та кваліфікованих майстрів, здатних виконати якісний ремонт застарілих систем. Відповідно, нові автомобілі коштують дорожче. Винятком є колекційні або лімітовані серії минулих років, вартість яких з часом може тільки зростати через їхню рідкість та історичну цінність.

Технічні характеристики – це комплекс параметрів, що описують будову та можливості автомобіля. Вони включають:

– Двигун: визначає вид палива (бензин, дизель, електрика, гібрид) та робочий об’єм (наприклад, 2.0 л).

– Потужність: виражається в кінських силах (к.с.) або кіловатах (кВт), відображаючи граничні можливості силової установки.

– Крутний момент: вимірюється в ньютон– метрах (Нм); від нього залежить тяга двигуна та інтенсивність прискорення.

– Трансмісія: тип коробки передач (механічна, автоматична або роботизована з електронним управлінням).

– Паливна система: вказує на пальне (газ, бензин, дизель, електроенергія) та спосіб його подачі.

– Розміри: довжина, висота, ширина та маса авто, що впливають на простір у салоні, маневреність та стійкість.

– Підвіска: тип конструкції (незалежна, пневматична, залежна тощо), від якої залежить плавність ходу та керованість.

– Динаміка та швидкість: час розгону до 100 км/год та максимально можлива швидкість руху.

– Тип приводу: передній, задній або повний, що визначає поведінку машини на дорозі та її прохідність.

– Витрата палива: середній показник споживання пального в різних режимах (місто/траса), що є ключовим для оцінки економічності.

– Тип кузова: форма авто (кросовер, седан, вантажівка, купе тощо), яка визначає його функціональне призначення.

– Системи безпеки: наявність допоміжних технологій, таких як подушки безпеки, ESP, адаптивний круїз–контроль, паркувальні асистенти та системи екстреного гальмування.

Сукупність цих даних дозволяє потенційному власнику оцінити реальну вартість транспортного засобу, де пріоритетність кожної характеристики залежить від особистих вимог.

Додатково під час оцінювання та прогнозування ціни варто зважати на загальний стан, пробіг та історію володіння. Стан автомобіля відображає його візуальний вигляд, справність агрегатів та ступінь зношеності компонентів. Пробіг (відстань у кілометрах чи милях) свідчить про інтенсивність експлуатації: зазвичай менший пробіг є ознакою кращого збереження ресурсу машини. Кількість попередніх власників також має значення, оскільки це впливає на прозорість історії обслуговування. Автомобілі, які належали одній людині, часто вважаються привабливішими, бо простіше відстежити, як за ними доглядали.

Підсумовуючи, ґрунтовне знання базових понять автомобільної галузі є необхідним для адекватної оцінки вартості машини. Аналіз усіх перелічених параметрів допомагає приймати виважені рішення під час купівлі, продажу, страхування або отримання фінансування на придбання авто.

1.3 Аналіз існуючих аналогів

У сучасних умовах глобальної цифровізації процеси прийняття рішень на автомобільному ринку дедалі більше спираються на автоматизовані аналітичні інструменти. Висока динамічність вторинного ринку та велика кількість змінних параметрів кожного окремого транспортного засобу зумовили появу низки потужних веб–ресурсів, які спеціалізуються на предиктивному аналізі цін.

Згідно з актуальними дослідженнями провідних автомобільних експертних платформ, до списку найкращих світових сервісів для оцінки та купівлі вживаних автомобілів входять такі ресурси, як CarGurus, Autotrader, Cars.com, Edmunds та Carvana [3]. Кожен із цих сервісів використовує власні інтелектуальні алгоритми:

- CarGurus здобув популярність завдяки системі «Instant Market Value» (IMV), яка в режимі реального часу аналізує мільйони оголошень для визначення того, наскільки запропонована ціна відповідає середньоринковій.
- Autotrader та Edmunds фокусуються на наданні максимально деталізованих звітів щодо оцінки вартості (Trade–In Value), базуючись на комбінації ринкових тенденцій та історичних даних про фактичні транзакції [3].

Математичним фундаментом більшості цих платформ є регресійний аналіз. Як зазначається у науково–методичних джерелах, регресійний аналіз – це статистичний метод дослідження, який дозволяє кількісно визначити вплив однієї або декількох незалежних змінних (таких як рік випуску, пробіг, об’єм двигуна) на залежну змінну, якою у даному випадку є ціна автомобіля [4]. Саме завдяки математичному моделюванню взаємозв’язків між цими факторами стає можливим не лише прогнозування конкретної вартості, а й розуміння ступеня впевненості у кожному з факторів.

Попри високу ефективність та мільйонні аудиторії зазначених аналогів, детальне вивчення їхньої специфіки виявляє низку критичних обмежень, які роблять розробку власної системи не просто альтернативою, а необхідним кроком для вирішення конкретних прикладних задач.

По–перше – проблема регіональної детермінованості та «інформаційного вакууму» на локальних ринках. Більшість топових ресурсів, таких як Edmunds чи Carvana, орієнтовані виключно на ринки США та Західної Європи [4]. Їхні моделі навчені на специфічних даних цих регіонів, де структура ціноутворення залежить від податкових пільг, вартості страхування та логістичних витрат, що є абсолютно нерелевантними для інших ринків. Наприклад, модель, що успішно прогнозує ціну в Каліфорнії, продемонструє критичну похибку при спробі оцінити авто на ринках Індії чи України через різницю у пріоритетності брендів, типах палива (LPG, дизель) та комплектаціях. Власна розробка, що базується на регіонально орієнтованому датасеті, дозволяє врахувати унікальні ознаки, такі як «тип власника» або специфічні одиниці вимірювання потужності та крутного моменту, які часто ігноруються глобальними платформами.

По–друге – відсутність прозорості та інтерпретованості алгоритмів. Комерційні аналоги надають користувачу лише фінальну цифру – прогноз вартості. Проте для наукового дослідження та професійної аналітики важливо розуміти структуру моделі. Згідно з засадами статистичного аналізу, регресійний метод дає змогу визначити, які змінні дійсно мають вплив на результат, а які є статистично незначущими та можуть бути ігноровані [4]. Глобальні сервіси приховують ці ваги як комерційну таємницю. Власна система, навпаки, забезпечує повну прозорість: розробник може наочно продемонструвати, наскільки кожна одиниця пробігу або кожен рік віку авто зменшує його вартість. Це створює додаткову цінність для користувача, який отримує не просто число, а обґрунтовану аналітику.

По–третє – економічна та технологічна бар'єрність. Професійні інструменти оцінки від CarGurus або Autotrader часто інтегровані в платні

екосистеми для дилерів. Для індивідуального користувача або малого бізнесу доступ до глибокої аналітики цих сервісів закритий. Створення власного програмного рішення на базі відкритої мови програмування Python та бібліотеки Streamlit дозволяє демократизувати доступ до складних методів машинного навчання. Така система не потребує складного встановлення, є кросплатформеною та може бути розгорнута на локальних серверах без необхідності сплачувати за дорогі підписки на закордонні API.

Аргументація переваги власної розробки:

1. Гнучкість навчання: На відміну від статичних сайтів–аналогів, архітектура власного проєкту дозволяє миттєво «перенавчити» модель на нових даних у разі різкої зміни ринкової ситуації (наприклад, стрибок курсу валют або зміна митних правил). Глобальні аналоги реагують на такі локальні зміни з великою затримкою.
2. Адаптація вхідних параметрів: Власна розробка дозволяє експериментувати з ознаками. Ми можемо включати або виключати параметри на кшталт «torque» (крутний момент) або «seller_type», спостерігаючи за зміною точності, що неможливо зробити в закритих інтерфейсах Edmunds чи CarGurus.
3. Практична спрямованість: Використання Streamlit перетворює теоретичне дослідження на готовий програмний продукт, який можна використовувати як мобільний додаток для швидкої оцінки авто «на місці», що значно підвищує зручність для кінцевого користувача.

Таким чином, розроблення власної інформаційної системи є науково обґрунтованим кроком. Це не просто копіювання існуючих функцій, а створення спеціалізованого, прозорого та регіонально адаптованого інструменту, який за рахунок методів регресійного аналізу забезпечує високу точність прогнозування у специфічних ринкових сегментах, недоступних для глобальних аналогів.

1.4 Огляд методів машинного навчання для вирішення задачі прогнозування

Через високу динамічність авторинку, складну структуру цінових факторів та потребу обробляти значні масиви даних найбільш перспективним інструментом для оцінки вартості є машинне навчання (Machine Learning, ML) - підмножина штучного інтелекту, що дозволяє системі самостійно виявляти приховані закономірності між характеристиками автомобіля та його ціною без жорсткого програмування кожного кроку [5]. Оскільки в задачі наявні і вхідні характеристики, і відомий результат (ціна), вона належить до парадигми навчання з учителем (Supervised Learning) і формалізується як задача регресії [6]. Серед основних груп методів для її розв'язання розглянуто лінійну регресію (швидку та повністю інтерпретовану, але обмежену в описі нелінійностей), ансамблеві методи на основі дерев рішень - випадковий ліс і градієнтний бустинг (стійкі до аномалій, здатні вловлювати нелінійні залежності та зберігати можливість аналізу важливості ознак), а також штучні нейронні мережі (потужні, але менш прозорі та схильні до перенавчання на малих вибірках). З огляду на потребу поєднати високу точність із прозорістю оцінки, у роботі зроблено акцент на регресійних та ансамблевих моделях, які детально розглянуто у другому розділі.

1.5 Постановка задачі

На основі проведеного аналізу предметної області, систематизації понятійного апарату автомобільної галузі та огляду існуючих методів машинного навчання сформульовано постановку задачі дослідження.

Метою роботи є розроблення інформаційної системи, яка на основі технічних та експлуатаційних характеристик автомобіля прогнозує його орієнтовну ринкову вартість із використанням методів машинного навчання. З формальної точки зору задача належить до класу задач регресії в межах

парадигми навчання з учителем (Supervised Learning): за вектором вхідних ознак автомобіля необхідно передбачити неперервне числове значення – ціну.

Вхідними даними для системи є набір характеристик автомобіля, що складається з десяти ознак: марка (Brand), модель (Model), рік випуску (Year), об'єм двигуна (Engine_Size), тип палива (Fuel_Type), тип коробки передач (Transmission), пробіг (Mileage), кількість дверей (Doors), кількість попередніх власників (Owner_Count) та ціна (Price), яка виступає цільовою змінною. Як навчальний матеріал використано набір даних обсягом 10 000 записів, що містить як числові, так і категоріальні ознаки.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- провести розвідувальний аналіз даних (EDA): дослідити розподіли ознак, виявити залежності між характеристиками автомобіля та його ціною, побудувати кореляційну матрицю та проаналізувати наявність викидів і пропущених значень;
- виконати попередню обробку даних: кодування категоріальних ознак методом One–Hot Encoding, групування неперервних ознак (об'єму двигуна, пробігу та року випуску) у відповідні діапазони, а також масштабування числових ознак за допомогою RobustScaler, стійкого до впливу викидів;
- побудувати та порівняти низку регресійних моделей машинного навчання – лінійну регресію, регресію з регуляризацією (Ridge, Lasso, ElasticNet), метод k– найближчих сусідів, дерево рішень, випадковий ліс, градієнтний бустинг, а також сучасні бібліотеки бустингу XGBoost, LightGBM та CatBoost;
- здійснити оптимізацію гіперпараметрів моделей із застосуванням методу пошуку по сітці (GridSearchCV) та крос–валідації;
- оцінити якість прогнозування за метриками середньої абсолютної похибки (MAE), середньоквадратичної похибки (MSE), кореня із середньоквадратичної похибки (RMSE) та коефіцієнта детермінації (R^2), на підставі чого обрати модель з найкращими показниками;

– реалізувати програмний застосунок із графічним інтерфейсом на основі бібліотеки Streamlit, який дозволяє користувачеві вводити характеристики автомобіля та отримувати прогноз його вартості в інтерактивному режимі.

Критерієм успішності розв'язання задачі є досягнення високої точності прогнозу, що підтверджується значенням коефіцієнта детермінації R^2, близьким до одиниці, та мінімальними значеннями похибок MAE й RMSE на тестовій вибірці.

Результатом виконання поставлених завдань має стати працездатна інформаційна система, здатна об'єктивно оцінювати ринкову вартість автомобіля на основі його характеристик, що дозволяє мінімізувати вплив суб'єктивного людського фактора та скоротити час на оцінювання.

2 МЕТОДИ ТА МЕТРИКИ ПРОГНОЗУВАННЯ ВАРТОСТІ АВТОМОБІЛІВ

2.1 Огляд алгоритмів регресії

Як було показано у першому розділі, визначення ринкової вартості автомобіля формалізується як задача регресії – відновлення неперервної залежності цільової змінної (ціни) від набору вхідних ознак. Кожен автомобіль набору даних описується вектором ознак, що поєднує числові характеристики (рік випуску, об’єм двигуна, пробіг) та категоріальні (марка, модель, тип палива, тип коробки передач, кількість дверей, кількість попередніх власників). Цільовою змінною є ціна, виражена у доларах США.

Формально завдання зводиться до пошуку функції, яка за вектором ознак конкретного автомобіля повертає прогнозовану ціну, мінімізуючи розбіжність між прогнозом і реальним значенням на навчальній вибірці. Найпростішою такою функцією є лінійна регресійна модель, загальний вигляд якої подано формулою (2.1):

$$Y = \omega_0 + \sum_{i=1}^{n} \omega_i x_i + \varepsilon, \qquad (2.1)$$

де Y - прогнозована ціна автомобіля;

$x_1 \dots x_n$ - значення вхідних ознак;

ω_0 - вільний член (зміщення);

$\omega_1 \dots \omega_n$ - вагові коефіцієнти, що визначають внесок кожної ознаки;

ε - випадкова похибка, яка враховує вплив неврахованих чинників

Лінійна модель є зручним базовим орієнтиром завдяки своїй прозорості, проте залежність ціни від ознак на вторинному ринку не завжди лінійна. Тому поряд із лінійними методами в роботі застосовано ансамблеві моделі на основі дерев рішень, здатні автоматично вловлювати нелінійні зв’язки та взаємодії ознак. Завдання обчислювального експерименту полягає не лише в навчанні

окремої моделі, а й у порівнянні кількох принципово різних підходів, щоб обрати той, який дає найточніший прогноз на незалежній тестовій вибірці.

2.2 Огляд застосованих методів машинного навчання

Для всебічного порівняння реалізовано одинадцять регресійних алгоритмів, які умовно поділяються на три групи: лінійні моделі, метричні та деревоподібні методи, а також ансамблі градієнтного бустингу. Такий набір охоплює весь спектр поширених підходів – від найпростіших і легко інтерпретованих до складних ансамблевих, що традиційно демонструють найвищу точність на табличних даних. Усі класичні алгоритми взято з бібліотеки scikit– learn [7], а сучасні методи бустингу – зі спеціалізованих бібліотек XGBoost, LightGBM і CatBoost.

2.2.1 Лінійні моделі. Лінійна регресія (Linear Regression, LR) відновлює пряму залежність ціни від ознак шляхом мінімізації суми квадратів відхилень. Її головна перевага – повна інтерпретованість: кожен коефіцієнт прямо показує, як зміна відповідної ознаки впливає на прогнозовану ціну. Водночас лінійна модель має обмежену здатність описувати складні нелінійні процеси, характерні для реального авторинку.

Щоб зменшити ризик перенавчання за наявності корельованих ознак, застосовують регуляризовані версії лінійної моделі. Гребенева регресія (Ridge) доповнює базову модель L2–регуляризацією, яка штрафує великі значення коефіцієнтів. Регресія Lasso використовує L1-регуляризацію, що здатна обнуляти коефіцієнти найменш значущих ознак і фактично виконує їх автоматичний відбір. Метод еластичної сітки (ElasticNet) поєднує обидва типи регуляризації, керуючи їх співвідношенням через окремий параметр. Реалізації всіх чотирьох моделей надає бібліотека scikit– learn [7].

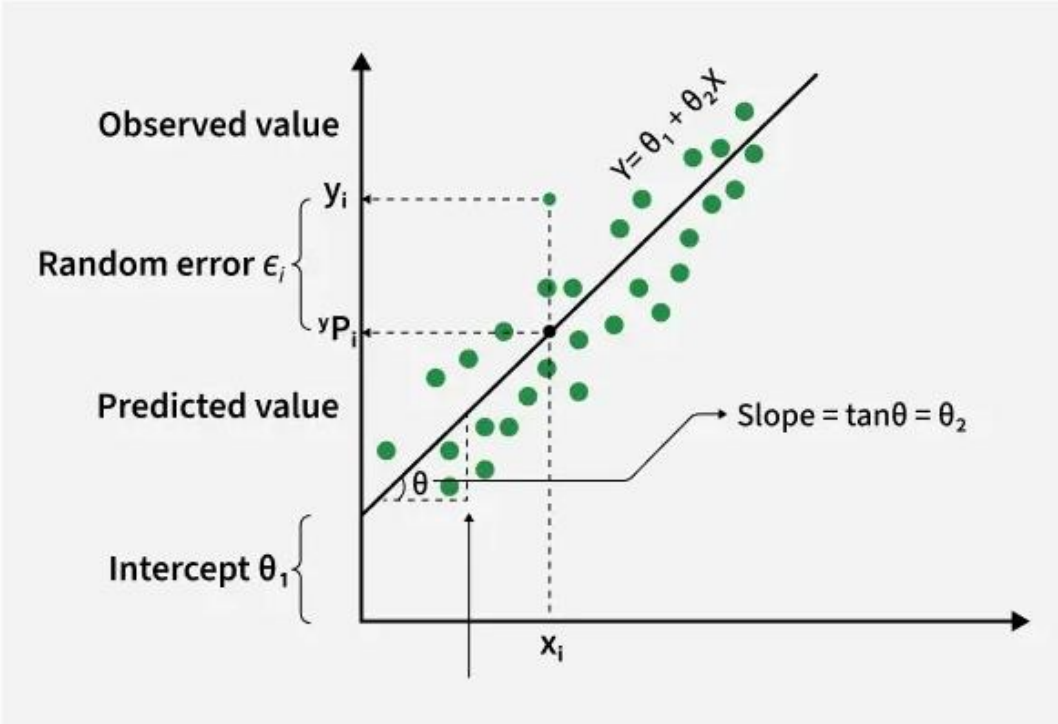

Рисунок 2.1 – Геометрична інтерпретація лінійної регресії

Геометрично робота лінійної регресії полягає у проведенні через хмару точок прямої (а у випадку кількох ознак гіперплощини), яка найкраще описує залежність цільової змінної від предикторів [17]. Метод оцінює коефіцієнти лінійного рівняння так, щоб мінімізувати розбіжності між прогнозованими та фактичними значеннями, тобто будує лінію найкращого наближення за критерієм найменших квадратів. Саме завдяки простоті та прозорості такої моделі кожен її коефіцієнт має зрозумілу інтерпретацію, що робить лінійну регресію зручною відправною точкою для прогнозування ціни автомобіля.

2.2.2 Метричні та деревоподібні методи. Метод k найближчих сусідів (KNN) прогнозує ціну автомобіля як усереднене значення цін найбільш схожих на нього об’єктів навчальної вибірки. Він простий концептуально, проте чутливий до масштабу ознак та їх кількості, що обмежує його точність на багатовимірних даних.

Дерево рішень (CART) будує ієрархію логічних правил, послідовно розбиваючи вибірку за значеннями ознак. Воно добре вловлює нелінійності, але окреме дерево схильне до перенавчання. Цей недолік усуває випадковий ліс (Random Forest, RF) – ансамблевий метод, запропонований Л. Брейманом

[8], який поєднує прогнози багатьох дерев, навчених на різних випадкових підвибірках даних та ознак. Усереднення результатів окремих дерев підвищує стійкість моделі до аномалій і знижує дисперсію прогнозу. Узагальнену схему роботи випадкового лісу подано на рисунку 2.2.

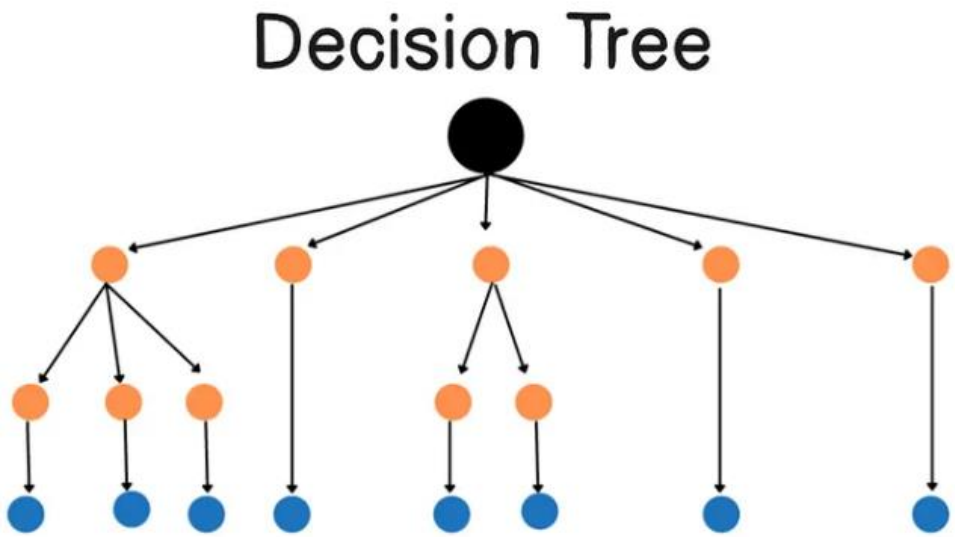

Рисунок 2.2 – Загальна структура дерева рішень

Дерево рішень має ієрархічну структуру: починаючи з кореневого вузла, дані послідовно розбиваються у внутрішніх вузлах за значеннями ознак, аж поки не досягаються листя, що містять підсумковий прогноз. На кожному кроці ознака для розбиття обирається так, щоб максимально зменшити неоднорідність підвибірок; для задач регресії як міру неоднорідності використовують дисперсію цільової змінної [18]. Перевагою дерева є здатність вловлювати нелінійні залежності без масштабування ознак, проте окреме глибоке дерево схильне до перенавчання, що й мотивує перехід до ансамблевих методів.

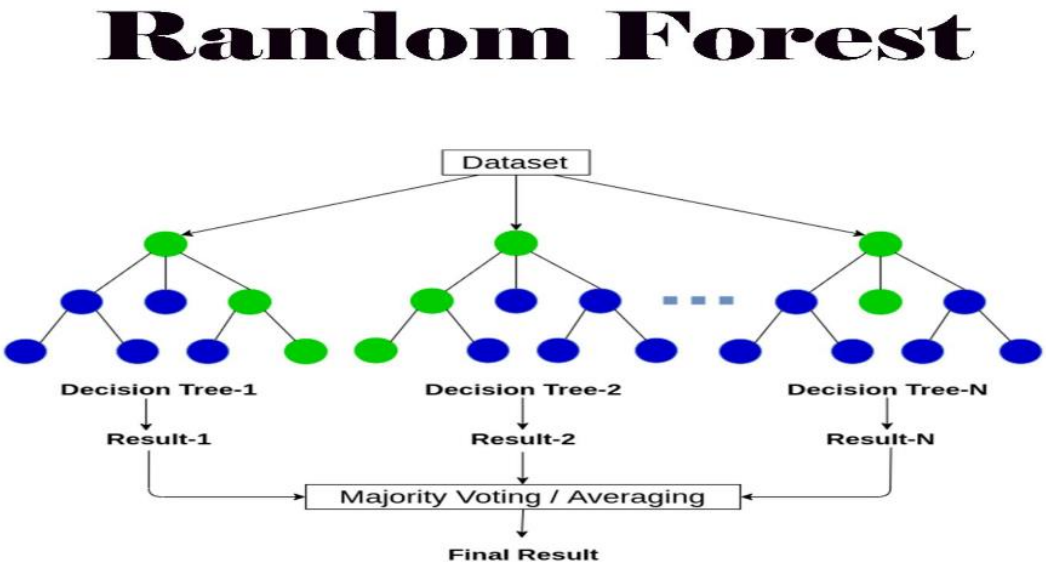

Рисунок 2.3 – Схема роботи алгоритму випадкового лісу

На першому етапі формується набір дерев на основі випадкових підмножин навчальних даних, на другому – кожне дерево дає власний прогноз, а підсумковий результат регресії обчислюється як усереднення цих прогнозів. Така архітектура поєднує гнучкість окремого дерева зі стійкістю ансамблю [19]. Ще вищу точність на табличних даних зазвичай забезпечують методи градієнтного бустингу, розглянуті далі.

2.2.3 Методи градієнтного бустингу. Окрему групу складають ансамблі градієнтного бустингу, у яких дерева додаються послідовно, і кожне наступне дерево виправляє помилки попередніх, рухаючись у напрямку антиградієнта функції втрат [16]. У роботі застосовано чотири реалізації цього підходу: класичний градієнтний бустинг (GBM) з бібліотеки scikit–learn [7] та три сучасні спеціалізовані бібліотеки. XGBoost вирізняється масштабованістю й регуляризацією, що контролює складність моделі [9].

LightGBM прискорює навчання за рахунок гістограмного пошуку розбиттів та вирощування дерев за листками [10]. CatBoost застосовує впорядкований бустинг і власний механізм обробки категоріальних ознак, що знижує зміщення прогнозу [11]. Попри відмінності в реалізації, ці методи

об’єднані спільною ідеєю поступового зменшення похибки, і саме вони, як засвідчив експеримент, дали найкращі результати.

Перехід від опису методів до їх кількісного порівняння потребує чітких критеріїв якості, які розглянуто в наступному підрозділі.

2.3 Вибір архітектури додатку

Для об’єктивного порівняння моделей застосовано чотири стандартні метрики регресії, реалізовані в модулі metrics бібліотеки scikit– learn [7]. Їх спільне використання дозволяє оцінити точність прогнозу з різних боків: як середню величину помилки, як чутливість до великих відхилень і як частку поясненої дисперсії.

Середня абсолютна похибка (MAE) обчислюється як середнє арифметичне абсолютних відхилень прогнозу від реального значення. Вона виражається в тих самих одиницях, що й ціна, і легко інтерпретується: показує, на скільки доларів у середньому помиляється модель. Перевагою MAE є стійкість до окремих аномальних значень.

$$\mathrm{M}AE = \frac{1}{m}\sum_{j=1}^{m}\left|y_j - \hat{y}_j\right|, \qquad (2.2)$$

де m - кількість об’єктів вибірки;

y_j - реальна ціна j-го автомобіля, USD;

$\hat{y}_j$ - прогнозована ціна j-го автомобіля, USD.

Середньоквадратична похибка (MSE) усереднює квадрати відхилень і тому сильніше штрафує великі помилки (формула 2.3).

$$MSE = \frac{1m^m}{\sum_{j=1}} (y_j - \hat{y}_j)^2, \qquad (2.3)$$

де m - кількість об’єктів вибірки;

y_j - реальна ціна j-го автомобіля, USD;

$\hat{y}_j$ - прогнозована ціна j-го автомобіля, USD.

Корінь із середньоквадратичної похибки (RMSE) повертає метрику до одиниць ціни й часто використовується як основний показник при оптимізації (формула 2.4).

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(y_j - \hat{y}_j)^2}, \quad (2.4)$$

де m — кількість об’єктів вибірки;

y_j — реальна ціна j-го автомобіля, USD;

$\hat{y}_j$ — прогнозована ціна j-го автомобіля, USD.

Коефіцієнт детермінації (R^2) показує, яку частку розкиду цін модель пояснює (формула 2.5). Його значення лежить у межах від нуля до одиниці: чим ближче до одиниці, тим точніша модель. Саме за цією метрикою у роботі обирається найкраща модель, оскільки вона дає узагальнену оцінку якості незалежно від масштабу цін.

$$R^2 = 1 - \frac{\sum_{j=1}^{m}\left(y_j - \hat{y}_j\right)^2}{\sum_{j=1}^{m}\left(y_j - \bar{y}\right)^2}, \quad (2.5)$$

де yi – реальна ціна i – го автомобіля;

ŷi – прогнозована ціна;

ȳ – середня ціна по вибірці;

n – кількість об’єктів вибірки.

2.4 Проєктування інтерфейсу користувача та налаштування гіперпараметрів

Спершу виконується розмежування ознак на числові та категоріальні за допомогою функції grab_col_names, яка автоматично визначає тип кожної колонки на основі типу даних та кількості унікальних значень.

На етапі конструювання ознак три числові характеристики розбито на інтервальні групи методом pd.cut: об’єм двигуна — на категорії Small, Medium і Large; пробіг - на Low, Medium і High; рік випуску - на періоди 2000s, 2010s і 2020s. Таке групування додає моделі узагальнену інформацію про клас автомобіля поряд із точними числовими значеннями.

Категоріальні ознаки кодуються методом One-Hot Encoding із параметром drop_first=True. Цей параметр відкидає по одній базовій категорії в кожній ознаці, що дозволяє уникнути мультиколінеарності (так званої пастки фіктивних змінних). Базовими (опущеними) категоріями виступають: для типу палива - Diesel, для кількості власників - 1, для марки - Audi, для коробки передач - Automatic, для кількості дверей - 2, а для груп об’єму, пробігу та року - відповідно Small, Low і 2000s. Після кодування набір ознак розширюється до 59 колонок.

Завершальним кроком передобробки є масштабування. Числові ознаки мають суттєво різні діапазони (рік випуску вимірюється тисячами, пробіг - сотнями тисяч), тому до трьох ознак — року, об’єму двигуна та пробігу - застосовано RobustScaler з бібліотеки scikit-learn [7]. На відміну від стандартного масштабування, він спирається на медіану та міжквартильний розмах і тому стійкий до викидів. Категоріальні ознаки після One-Hot кодування вже подані нулями та одиницями й масштабування не потребують.

Після базового порівняння моделей з параметрами за замовчуванням проведено налаштування гіперпараметрів методом повного перебору за сіткою GridSearchCV із п’ятиразовою перехресною перевіркою [7]. Для кожної моделі задано власну сітку значень: для гребеневої та Lasso-регресії перебиралися значення коефіцієнта регуляризації, для методу k сусідів - кількість сусідів, для деревоподібних і бустингових моделей - глибина дерев, кількість дерев та темп навчання. Перехресна перевірка розбиває навчальну вибірку на п’ять частин і по черзі використовує кожну з них для контролю, що дає надійнішу оцінку якості, ніж одноразове розбиття. Найкраща модель обирається за коефіцієнтом детермінації R^2 на тестовій вибірці.

Описані у цьому розділі методи, метрики та процедури передобробки повністю реалізовано програмно. Конкретні засоби реалізації, результати обчислювального експерименту та розроблений застосунок розглянуто в наступному розділі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Засоби програмної реалізації

Вибір мови програмування є важливим для успіху будь–якого проєкту, оскільки безпосередньо впливає на швидкість розробки, доступність бібліотек і подальшу підтримуваність продукту. Систему реалізовано мовою Python – високорівневою інтерпретованою мовою з автоматичним управлінням пам’яттю, яка є фактичним стандартом у галузі науки про дані завдяки розвиненій екосистемі бібліотек.

Обробку та аналіз даних виконано засобами pandas [12] і NumPy [13], побудову графіків – за допомогою matplotlib [14] і seaborn [15]. Класичні алгоритми машинного навчання та інструменти оцінювання взято з бібліотеки scikit–learn [7], а сучасні методи градієнтного бустингу – з бібліотек XGBoost [9], LightGBM [10] і CatBoost [11]. Розробку та проведення обчислювального експерименту здійснено в середовищі Jupyter Notebook, яке поєднує код, його результати та пояснювальний текст в одному документі. Інтерактивний вебінтерфейс для кінцевого користувача побудовано на фреймворку Streamlit, що дає змогу швидко перетворити модель машинного навчання на повноцінний вебзастосунок без написання серверного коду. Навчена модель та допоміжні об’єкти зберігаються у форматі pickle.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import itertools
import plotly.graph_objects as go
import time

from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.exceptions import ConvergenceWarning
from sklearn.linear_model import LogisticRegression, LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split, cross_val_score,GridSearchCV
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, StandardScaler, RobustScaler

import warnings
warnings.simplefilter(action="ignore")
```

Рисунок 3.1 – Список бібліотек

Маючи визначений набір інструментів, перейдемо до характеристики даних, на яких навчалася система.

3.2. Опис набору даних

Інформаційним фундаментом для проєктування та навчання предиктивного ядра системи є відкритий набір даних «Car Prices Prediction Data», опублікований на платформі Kaggle [20]. Набір поширюється у форматі CSV і призначений саме для задач регресії та навчання предиктивних моделей ціноутворення. Загальний обсяг сформованої вибірки становить рівно 10 000 записів про пропозиції на автомобільному ринку. Структурна розмірність початкової матриці складається з 10 ознак, з яких 9 виступають як незалежні змінні (вхідні предиктори), а 1 - як цільова залежна величина (ціна).

За допомогою розроблених автоматизованих алгоритмів експрес–діагностики структур даних на початковому етапі було виконано сканування масиву з метою верифікації типів даних, визначення повноти заповнення та

первинного квантильного аналізу. Результати тестування підтвердили 100% заповненість набору даних: у жодній з колонок не виявлено пропущених значень або дефектів структури. Медіанне значення цільової змінної зафіксовано на рівні 9322 умовних одиниць, а медіанний пробіг становить 152 341 км, що відображає реалістичний стан об'єктів на вторинному ринку.

Для забезпечення можливості математичного моделювання всі ознаки було класифіковано за типами змінних. Повний системний опис кожної ознаки, її системний тип у середовищі розробки та граничні межі значень наведено в таблиці 3.1.

Таблиця 3.1 – Інженерно–технічна специфікація ознак набору даних

Назва ознаки	**Категорія змінної**	**Опис характеристики та діапазон значень**
Brand	Категоріальна	Марка автомобіля (10 марок)
Model	Категоріальна	Модель (30 моделей)
Year	Числова	Рік випуску (2000–2023)
Engine_Size	Числова	Об’єм двигуна, л (1.0–5.0)
Fuel_Type	Категоріальна	Тип палива (Petrol, Diesel, Electric, Hybrid)
Transmission	Категоріальна	Тип КПП (Manual, Automatic, Semi–Automatic)
Mileage	Числова	Пробіг, км (25–299 947)
Doors	Категоріальна	Кількість дверей (2–5)
Owner_Count	Категоріальна	Кількість попередніх власників (1–5)
Price	Цільова	Ринкова ціна, USD (2000–18 301)

Перевірка набору даних показала його високу якість: пропущені значення відсутні повністю, а перевірка на викиди за міжквартильним методом не виявила аномальних значень у числових ознаках. Це дозволяє звести етап очищення даних до мінімуму, а основну увагу приділити аналізу залежностей та конструюванню ознак.

```
#################### Shape ####################
(10000, 10)
#################### Types ####################
Brand               str
Model               str
Year              int64
Engine_Size     float64
Fuel_Type           str
Transmission        str
Mileage           int64
Doors             int64
Owner_Count       int64
Price             int64
dtype: object
#################### Head ####################
        Brand   Model  Year  Engine_Size Fuel_Type    Transmission  Mileage
0         Kia     Rio  2020        4.200    Diesel          Manual   289944
1   Chevrolet  Malibu  2012        2.000    Hybrid       Automatic     5356
2    Mercedes     GLA  2020        4.200    Diesel       Automatic   231440
3        Audi      Q5  2023        2.000  Electric          Manual   160971
4  Volkswagen    Golf  2003        2.600    Hybrid  Semi-Automatic   286618

   Doors  Owner_Count  Price
0      3            5   8501
1      2            3  12092
...
Mileage     284443.700 296474.030 299947.000
Doors            5.000      5.000      5.000
Owner_Count      5.000      5.000      5.000
Price        13981.050  15745.080  18301.000
```

Рисунок 3.2 – Результати автоматизованого аудиту структури та типів даних

Представлений системний розподіл даних свідчить про високу інформаційну збалансованість вибірки, що створює надійні умови для переходу до етапу розвідувального аналізу (EDA) та подальшого навчання математичних моделей.

3.3 Розвідувальний аналіз даних (EDA)

Перед безпосереднім навчанням моделей машинного навчання було проведено розвідувальний аналіз даних (Exploratory Data Analysis, EDA) – комплекс процедур, спрямованих на вивчення структури набору даних, статистичних характеристик ознак, виявлення закономірностей, аномалій і взаємозв'язків між змінними. Метою цього етапу є формування цілісного

уявлення про природу даних, що дозволяє обґрунтовано обрати методи попередньої обробки та підвищити якість подальшого прогнозування.

3.3.1 Визначення типів ознак. На першому кроці аналізу всі ознаки набору даних було класифіковано за їхньою природою на категоріальні та числові, оскільки ці дві групи потребують принципово різних підходів до візуалізації та обробки. Для автоматизованого розподілу розроблено функцію grab_col_names, яка відносить ознаку до категоріальних або числових на основі її типу даних та кількості унікальних значень. Числові змінні з невеликою кількістю унікальних значень (менше заданого порогу) розглядаються як категоріальні за змістом, тоді як текстові змінні з надмірно великою кількістю унікальних значень виділяються в окрему групу як такі, що мають високу кардинальність.

У результаті застосування функції встановлено, що набір даних містить категоріальні ознаки (марка, модель, тип палива, коробка передач, кількість дверей, кількість власників) та числові ознаки (рік випуску, об'єм двигуна, пробіг, ціна). Така класифікація стала основою для подальшого диференційованого аналізу кожної групи. На рисунку 3.3 вказуються результати автоматизованої класифікації ознак набору даних.

```
[8]  cat_cols, num_cols, cat_but_car,  num_but_cat = grab_col_names(df)

...  Observations: 10000
     Variables: 10
     cat_cols: 6
     num_cols: 4
     cat_but_car: 0
     num_but_cat: 2
```

Рисунок 3.3 – Результати автоматизованої класифікації ознак набору даних

3.3.2 Аналіз категоріальних ознак. Для дослідження категоріальних ознак розроблено функцію cat_summary, яка для кожної ознаки виводить розподіл значень за категоріями та їхню частку у відсотках від загального обсягу вибірки, а також будує стовпчикову діаграму (countplot). Це дозволяє

оцінити збалансованість категорій і виявити рідкісні значення, що трапляються лише в незначній частині спостережень.

Аналіз показав, що категорії більшості ознак розподілені відносно рівномірно, що свідчить про достатню репрезентативність вибірки та відсутність суттєвого дисбалансу класів. Результати аналізу розподілу категоріальних ознак наведено на рисунку 3.4.

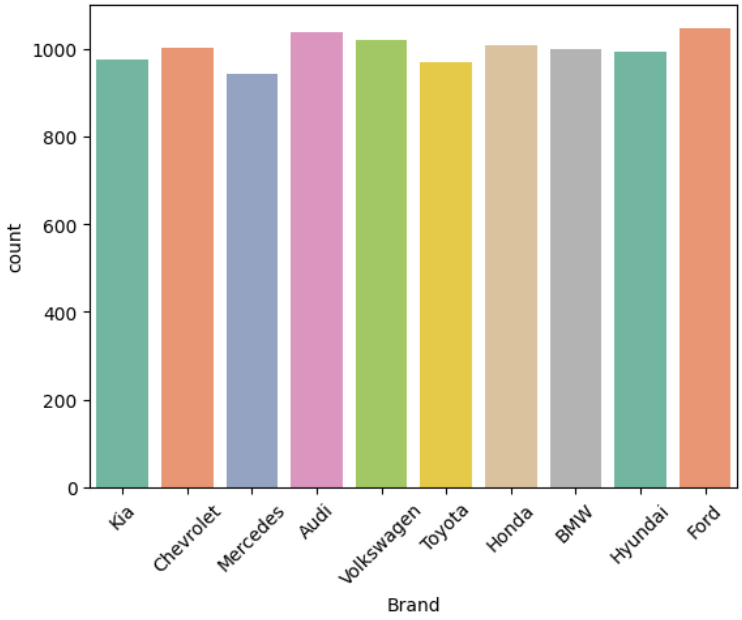

Рисунок 3.4 – Частотний розподіл автомобілів за маркою

На рисунку 3.4 представлено стовпчикову діаграму, яка відображає частотний розподіл транспортних засобів за маркою виробника (Brand) у досліджуваному наборі даних. Як видно з візуалізації, вибірка є репрезентативною та збалансованою в розрізі автомобільних брендів: кожна з десяти представлених марок (Kia, Chevrolet, Mercedes, Audi, Volkswagen, Toyota, Honda, BMW, Hyundai, Ford) містить приблизно рівну кількість спостережень (близько 1000 записів на кожну категорію). Такий рівномірний розподіл свідчить про відсутність дисбалансу класів за цією ознакою. Це є вкрай сприятливою умовою для подальшого навчання моделей машинного навчання, оскільки запобігає зміщенню алгоритмів (*bias*) у бік домінуючого бренду та гарантує об'єктивність прогнозування для всіх марок автомобілів.

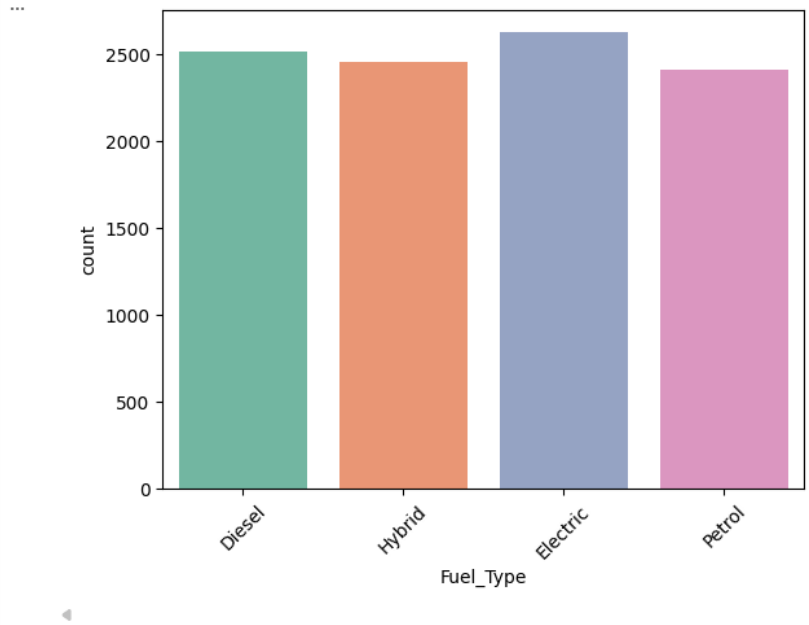

Рисунок 3.5 – Частотний розподіл транспортних засобів за типом палива

На рисунку 3.5 наведено результати аналізу категоріальної ознаки Fuel_Type (тип палива), що є одним із ключових факторів, які впливають на формування ринкової ціни автомобілів. Графічна візуалізація демонструє розподіл частот за чотирма основними категоріями: Diesel, Hybrid, Electric та Petrol. Як видно з діаграми, класи розподілені майже рівномірно, що підтверджує широке охоплення різних типів двигунів у досліджуваній вибірці. Така збалансованість даних за типом палива є важливою для коректної роботи моделей машинного навчання, оскільки дозволяє алгоритмам вивчити специфіку ціноутворення для кожного класу без переваги (зміщення) в бік одного з них.

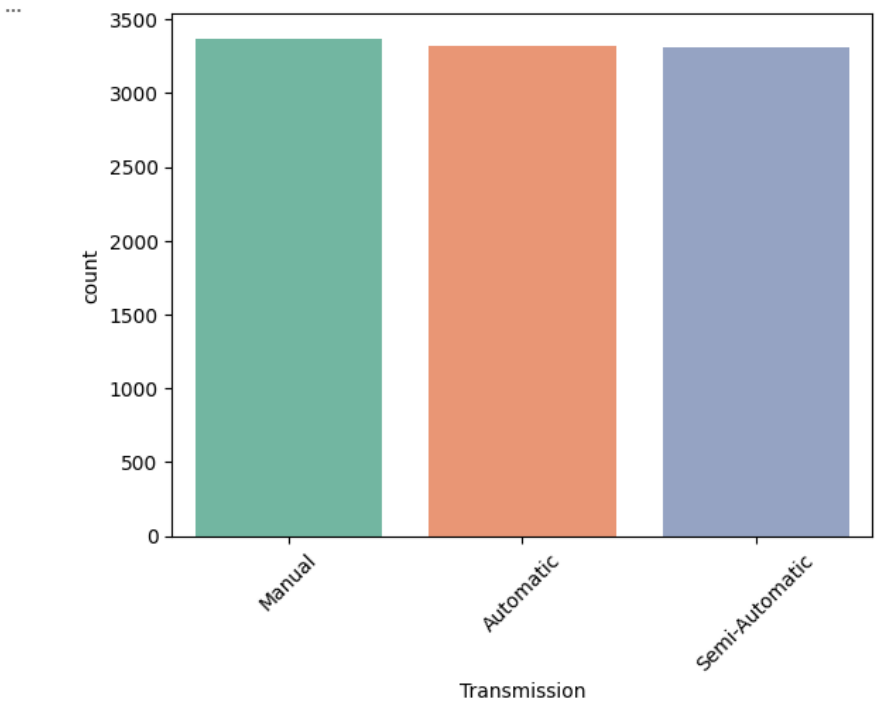

Рисунок 3.6 – Частотний розподіл транспортних засобів за типом трансмісії

На рисунку 3.6 представлена стовпчикова діаграма, що ілюструє розподіл транспортних засобів у вибірці за типом трансмісії (Transmission). Як свідчать результати візуалізації, набір даних демонструє високий ступінь збалансованості: частка автомобілів, оснащених механічною, автоматичною та напівавтоматичною коробками передач, є практично ідентичною і становить приблизно 33,3% для кожної категорії. Така рівномірність розподілу ознаки свідчить про об’єктивну представленість усіх технічних конфігурацій на ринку, що дозволяє уникнути зміщення прогнозних результатів під час навчання регресійних моделей і забезпечує їхню високу узагальнювальну здатність.

3.3.3 Аналіз числових ознак. Для числових ознак побудовано функцію num_summary, яка виводить розширену описову статистику (мінімум, максимум, середнє, стандартне відхилення та набір квантилів від 5 % до 99 %), а також гістограму розподілу з накладеною кривою щільності ймовірності (KDE). Аналіз квантилів дозволяє оцінити форму розподілу, ступінь його симетричності та наявність потенційних викидів у крайніх діапазонах значень.

За результатами аналізу встановлено характер розподілу таких ознак, як рік випуску, об'єм двигуна та пробіг.

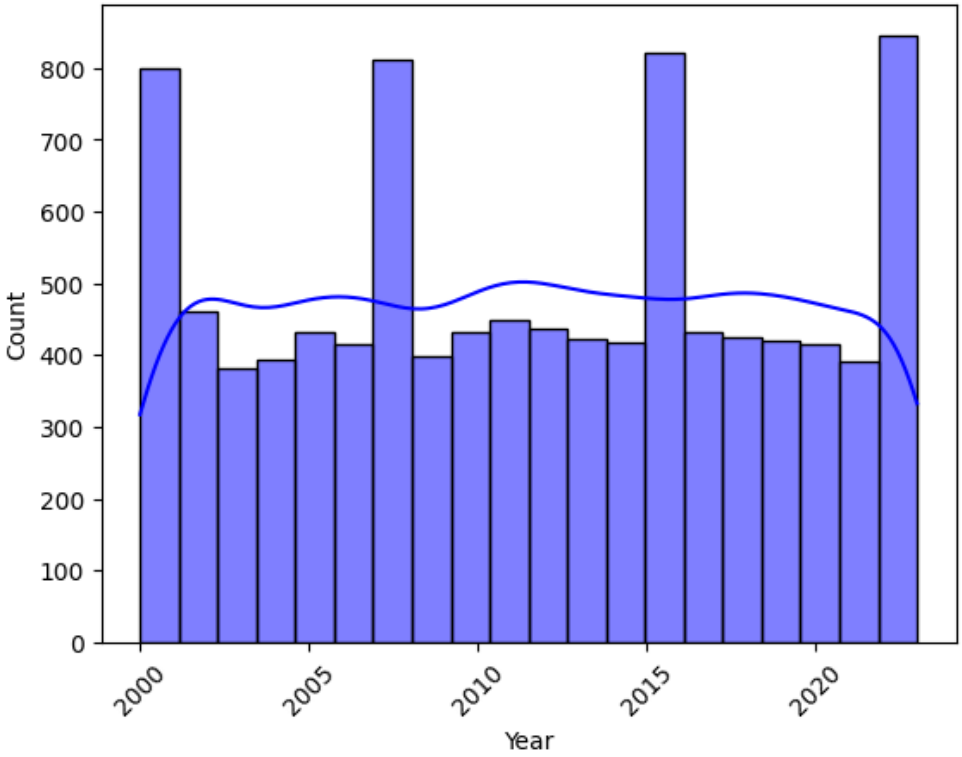

Рисунок 3.7 – Розподіл кількості автомобілів за роком випуску

Роки випуску охоплюють діапазон від 2000 до 2023 і розподілені відносно рівномірно. Наступна ознака – об’єм двигуна – показана на рисунку 3.4; вона зосереджена у проміжку від одного до п’яти літрів.

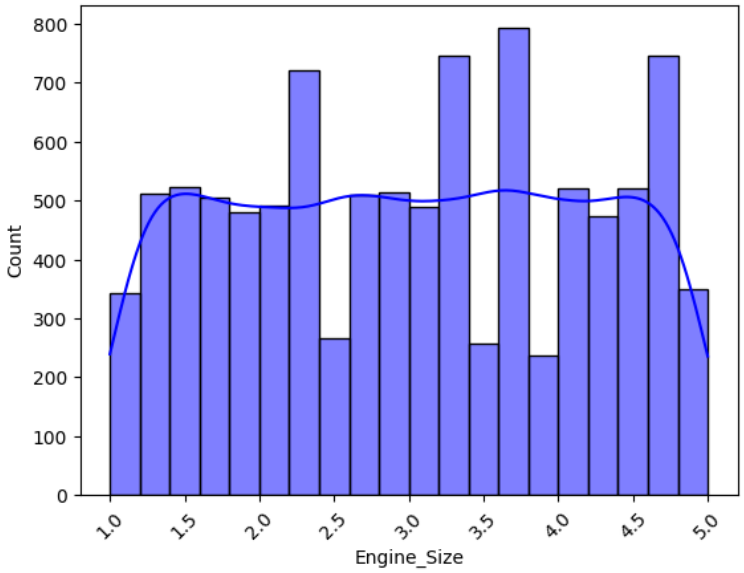

Рисунок 3.8 – Розподіл автомобілів за об'ємом двигуна

Найбільший практичний інтерес становить пробіг, адже він безпосередньо пов’язаний зі зношеністю автомобіля. Його розподіл наведено на рисунку 3.9.

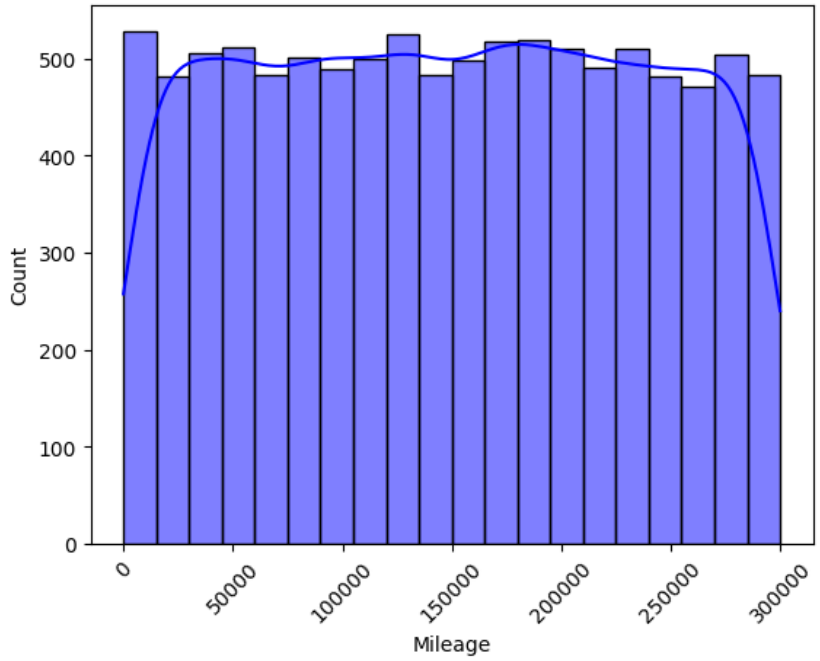

Рисунок 3.9 – Розподілу автомобілів за пробігом

Пробіг охоплює широкий діапазон – від кількох десятків кілометрів до майже трьохсот тисяч, що відповідає реальному ринку вживаних автомобілів різного ступеня використання. Зрозумівши, як розподілені окремі ознаки, перейдемо до головного питання – як вони пов’язані з ціною.

3.3.4 Залежність ціни від категоріальних ознак. Оскільки кінцевою метою роботи є прогнозування ціни автомобіля, окрему увагу приділено дослідженню впливу кожної ознаки на цільову змінну. Для категоріальних ознак за допомогою функції target_summary_with_cat обчислено середнє значення ціни в розрізі кожної категорії та побудовано відповідні стовпчикові діаграми.

Такий аналіз дозволяє наочно визначити, які категорії пов'язані з вищою або нижчою вартістю автомобіля. Зокрема, виявлено помітну різницю в середній ціні між автомобілями з різними типами палива, тоді як деякі ознаки (наприклад, кількість попередніх власників) демонструють майже однакову

середню ціну в усіх категоріях, що вказує на їхній незначний вплив на ціноутворення. Залежність середньої ціни від типу палива подано на рисунку 3.10.

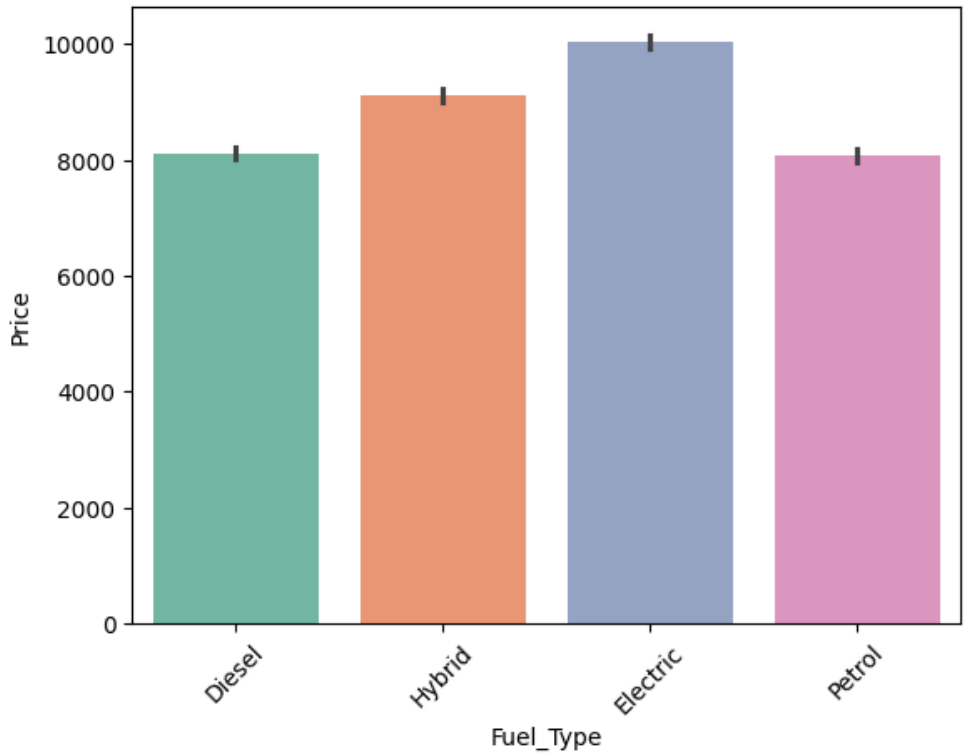

Рисунок 3.10 – Середня ціна автомобіля залежно від типу палива

Діаграма виявляє цікаву закономірність: бензинові та дизельні автомобілі коштують у середньому майже однаково (приблизно 8071 і 8117 доларів відповідно), тоді як гібридні (близько 9113 доларів) і особливо електричні (близько 10 032 доларів) є помітно дорожчими. Це відповідає ринковій логіці – електромобілі та гібриди як новіша технологія мають вищу вартість. На противагу цьому, залежність ціни від кількості попередніх власників, показана на рисунку 3.11, виглядає принципово інакше.

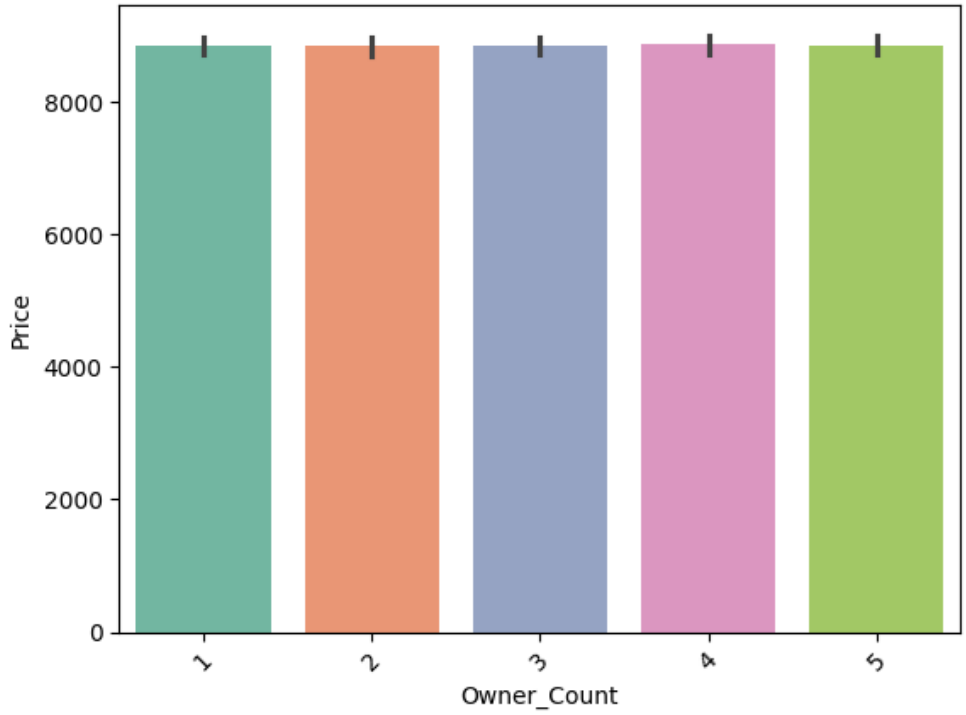

Рисунок 3.11 – Середня ціна автомобіля залежно від кількості власників

Середня ціна майже не змінюється зі зростанням кількості власників: для всіх п’яти груп вона коливається у вузькому коридорі близько 8840–8870 доларів. Попри теоретичні очікування (адже історія володіння традиційно вважається важливим чинником ціноутворення, що відзначалося в підрозділі 1.2), у цьому наборі даних кількість власників має слабкий вплив на ціну. Цей результат становить самостійний аналітичний висновок, який далі підтверджується кореляційним аналізом та аналізом важливості ознак. Водночас параметр збережено у складі моделі та в інтерфейсі застосунку, оскільки він є природною характеристикою автомобіля і може набути ваги на інших, більш репрезентативних наборах даних.

3.3.6 Кореляційний аналіз. Для кількісного оцінювання сили лінійних взаємозв'язків між числовими ознаками побудовано кореляційну матрицю з використанням функції high_correlated_cols2. Функція обчислює коефіцієнти кореляції Пірсона між усіма парами числових змінних, формує матрицю абсолютних значень та візуалізує її у вигляді теплової карти (heatmap).

Кореляційний аналіз виконує дві важливі функції. По–перше, він дозволяє виявити ознаки, що мають найбільший лінійний зв'язок із цільовою змінною, тобто є найбільш інформативними для прогнозування ціни.

По–друге, він дає змогу виявити явище мультиколінеарності – наявність сильно корельованих між собою ознак, що може негативно впливати на стійкість лінійних моделей. Теплову карту кореляцій наведено на рисунку 3.12. За результатами аналізу не виявлено пар ознак із надмірно високим коефіцієнтом кореляції, який перевищував би встановлений поріг, що свідчить про відсутність суттєвої мультиколінеарності в наборі даних.

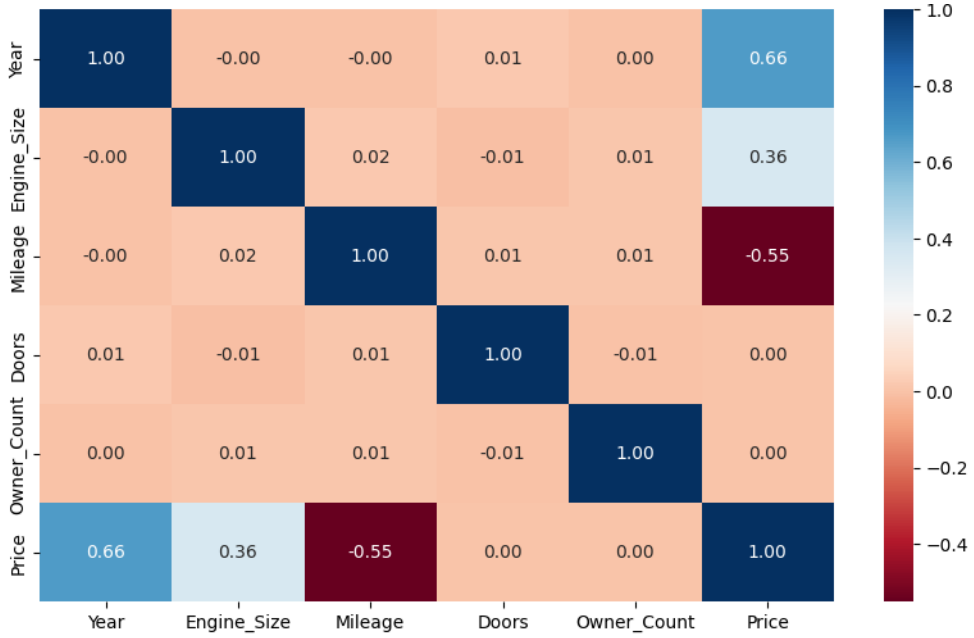

Рисунок 3.12 – теплова карта кореляцій числових ознак

Аналіз кореляцій із ціною дає чітку ієрархію впливу ознак. Найсильніший додатний зв’язок має рік випуску (коефіцієнт кореляції близько 0.663): новіші автомобілі очікувано дорожчі. Пробіг демонструє помітний від’ємний зв’язок (близько −0.551) – чим більший пробіг, тим нижча ціна. Об’єм двигуна корелює з ціною помірно додатно (близько 0.357). Натомість кількість дверей (близько 0.0005) і кількість власників (близько 0.0027) практично не корелюють із ціною, що узгоджується з висновком попереднього підрозділу про слабкий вплив історії володіння в цьому наборі даних.

3.3.7 Аналіз розподілу цільової змінної. Окремим важливим етапом є дослідження розподілу цільової змінної – ціни автомобіля. Для цього побудовано гістограму розподілу значень ціни. Аналіз показав, що розподіл ціни є асиметричним із правостороннім зміщенням (right–skewed): більшість автомобілів зосереджено в нижньому та середньому ціновому діапазоні, тоді як автомобілі високої вартості трапляються значно рідше й утворюють витягнутий «хвіст» розподілу. Розподіл ціни проаналізовано за допомогою гістограми, поданої на рисунку 3.13.

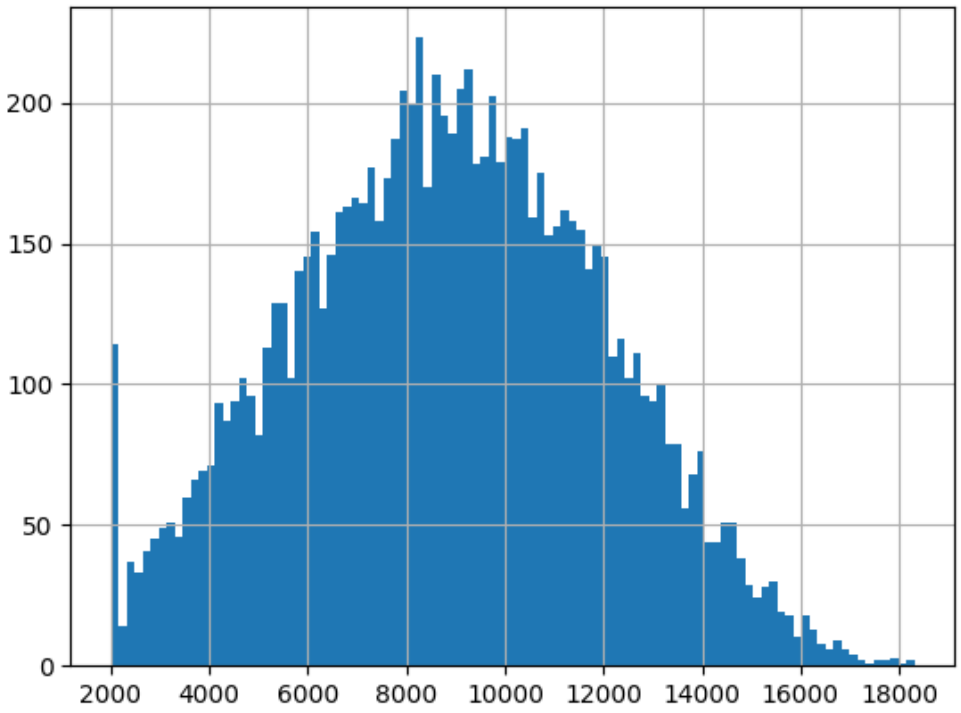

Рисунок 3.13 – Розподіл цільової змінної (ціни)

Оскільки асиметрія цільової змінної може погіршувати якість роботи регресійних моделей, додатково розглянуто логарифмічне перетворення ціни за формулою натурального логарифма від збільшеного на одиницю значення (log1p). Як видно з рисунка 3.14, логарифмоване значення ціни має розподіл, наближений до нормального, що є сприятливішою умовою для побудови моделей. Цей результат враховано на подальших етапах роботи.

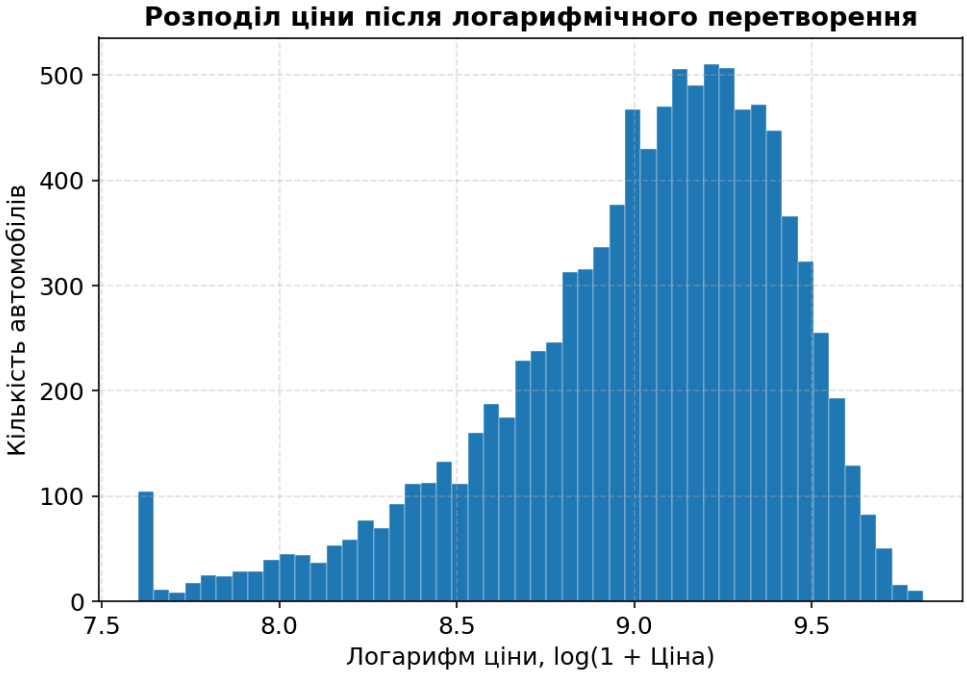

Рисунок 3.14 – Розподіл логарифмованої ціни автомобілів

Таким чином, проведений розвідувальний аналіз дозволив сформувати повне уявлення про структуру набору даних, оцінити статистичні характеристики та розподіли ознак, визначити характер їхнього впливу на ціну й виявити асиметрію цільової змінної. Одержані висновки стали підґрунтям для наступного етапу – попередньої обробки та інженерії ознак.

3.4 Передоробка ознак та навчння з порівнянням

Висновки розвідувального аналізу реалізовано програмно у вигляді послідовності перетворень. Спочатку три числові ознаки розбито на групи функцією pd.cut. Об’єм двигуна поділено на три інтервали з межами приблизно 2.33 і 3.67 літра (категорії Small, Medium, Large), пробіг – з межами близько 99 999 і 199 973 км (Low, Medium, High), а рік випуску – з межами 2007.7 і 2015.3 (періоди 2000s, 2010s, 2020s). Ці межі визначаються автоматично як рівні поділи діапазону кожної ознаки на три частини.

Далі всі категоріальні ознаки, включно з кількістю власників, закодовано методом One–Hot із параметром drop_first=True, після чого набір ознак

розширився до 59 колонок. Нарешті, до трьох числових ознак застосовано RobustScaler. Важлива деталь реалізації: масштабувальник навчається (fit) лише один раз і потім зберігається у файл, щоб під час роботи застосунку нові дані користувача масштабувалися точно за тими ж параметрами, що й навчальна вибірка.

	Brand	Model	Year	Engine_Size	Fuel_Type	Transmission	Mileage	Doors	Owner_Count	Price	Engine_Size_Group	Mileage_Group	Year_of_Registration_Group
0	Kia	Rio	2020	4.200	Diesel	Manual	289944	3	5	8501	Large	High	2020s
1	Chevrolet	Malibu	2012	2.000	Hybrid	Automatic	5356	2	3	12092	Small	Low	2010s
2	Mercedes	GLA	2020	4.200	Diesel	Automatic	231440	4	2	11171	Large	High	2020s
3	Audi	Q5	2023	2.000	Electric	Manual	160971	2	1	11780	Small	Medium	2020s
4	Volkswagen	Golf	2003	2.600	Hybrid	Semi-Automatic	286618	3	3	2867	Medium	High	2000s

Рисунок 3.15 – Виведено 5 рядків даних

Підготувавши ознаки, можна переходити до навчання та порівняння моделей.

Набір даних розділено на навчальну та тестову вибірки у співвідношенні 80 на 20 із фіксованим зерном генератора випадкових чисел (random_state=17), що гарантує відтворюваність результатів. Навчальна вибірка містить 8000 записів, тестова – 2000. На цих даних навчено всі одинадцять моделей із параметрами за замовчуванням, а їх якість оцінено за чотирма метриками. Зведені результати порівняння подано в таблиці 3.2.

Таблиця 3.2 – Порівняння точності регресійних моделей на тестовій вибірці

Модель	RMSE	R^2	MAE	MSE
LR	115.96	0.9986	23.72	13 446
Ridge	115.76	0.9986	25.20	13 399
Lasso	115.67	0.9986	26.40	13 380
ElasticNet	1691.36	0.7012	1365.03	2 860 705
KNN	1240.59	0.8392	993.20	1 539 064
CART	849.36	0.9246	636.81	721 420
RF	532.74	0.9704	433.43	283 814
GBM	229.07	0.9945	180.35	52 474
XGBoost	234.52	0.9943	185.29	54 998

Продовження таблиці 3.2

Модель	**RMSE**	**R^2**	**MAE**	**MSE**
LightGBM	180.55	0.9966	142.92	32 598
CatBoost	55.55	0.9997	31.22	3086

Результати порівняння дозволяють зробити кілька важливих висновків. Найвищий коефіцієнт детермінації показала модель CatBoost ($R^2 = 0.9997$ при RMSE близько 55.6 долара), тому саме її обрано як найкращу за основним критерієм. Дуже близькі за якістю результати дали інші бустингові моделі – LightGBM і GBM. Цікаво, що лінійні моделі (LR, Ridge, Lasso) також показали надзвичайно високу точність ($R^2 = 0.9986$) і навіть найменшу середню абсолютну похибку (для лінійної регресії MAE становить лише 23.7 долара). Помітно гірше впоралися ElasticNet і метод k сусідів, що пояснюється надмірною регуляризацією першого та чутливістю другого до масштабу й розмірності ознак.

Висока точність лінійних моделей не випадкова: розвідувальний аналіз показав переважно лінійний характер залежності ціни від ключових ознак (рік, пробіг, об’єм двигуна). Це робить лінійну модель привабливою з практичного погляду – вона проста, швидка та легко інтерпретується. Тому, попри формальну першість CatBoost за R^2, для розгортання у застосунку обрано модель Lasso як оптимальний компроміс між точністю, інтерпретованістю та компактністю збереженого файлу. Наочне порівняння моделей за метриками RMSE та R^2 подано на рисунку 3.16.

```
Hyperparameter Tuning for LR:
RMSE: 115.957 (LR)
R^2 Score: 0.9986 (LR)
MAE: 23.7184 (LR)
MSE: 13446.0301 (LR)
Execution Time: 0.01 seconds

Hyperparameter Tuning for Ridge:
Best parameters: {'alpha': 0.1}
RMSE: 115.9324 (Ridge)
R^2 Score: 0.9986 (Ridge)
MAE: 23.8521 (Ridge)
MSE: 13440.3318 (Ridge)
Execution Time: 2.79 seconds

Hyperparameter Tuning for Lasso:
Best parameters: {'alpha': 0.1}
RMSE: 115.8792 (Lasso)
R^2 Score: 0.9986 (Lasso)
MAE: 23.512 (Lasso)
MSE: 13427.9936 (Lasso)
Execution Time: 1.38 seconds

Hyperparameter Tuning for ElasticNet:
Best parameters: {'alpha': 0.1, 'l1_ratio': 0.9}
...
MSE: 74415.2958 (CatBoost)
Execution Time: 4.15 seconds

BEST MODEL: Lasso (R^2 = 0.9986)
```

Рисунок 3.16 – Порівняння моделей за метриками RMSE та R^2

Після відбору найкращих моделей проведено налаштування їх гіперпараметрів, що дозволило ще трохи підвищити точність. Якість фінальної моделі найкраще видно при безпосередньому порівнянні прогнозів із реальними цінами.

3.5 Аналіз точності прогнозів та збереження моделі

Для оцінки якості фінальної моделі на тестовій вибірці сформовано таблицю, що зіставляє прогнозовані та реальні ціни з обчисленням різниці між ними. Прогнози моделі відрізняються від реальних цін лише на десятки, рідше – на кількасот доларів. Наприклад, для автомобіля з реальною ціною 10 308

доларів модель спрогнозувала 10 503 долари, а для автомобіля вартістю 5447 доларів – 5432 долари. Кілька характерних прикладів зведено в таблиці 3.3.

Таблиця 3.3 – Приклади зіставлення прогнозованих і реальних цін

Прогноз, USD	**Реальна ціна, USD**	**Різниця, USD**
10320.297	10308	– 12.297
5470.990	5447	– 23.990
8726.907	8732	5.093
7878.234	7871	– 7.234
9143.260	9144	0.740

Такий рівень точності для більшості об’єктів є дуже високим: відносна похибка прогнозу зазвичай не перевищує кількох відсотків від вартості автомобіля. Це підтверджує, що обрані ознаки та модель добре описують механізм ціноутворення в наборі даних.

	Predicted Price	True Price	Difference
2688	10320.297	10308	-12.297
233	5470.990	5447	-23.990
9099	8726.907	8732	5.093
8652	7878.234	7871	-7.234
2842	9143.260	9144	0.740
...	...	...	...
4395	11810.385	11813	2.615
8669	4488.753	4465	-23.753
7634	12625.669	12629	3.331
556	2484.280	2445	-39.280
8247	11200.842	11192	-8.842

[2000 rows x 3 columns]

Рисунок 3.17 – Візуалізація прогнозних та фактичних значень цільової змінної

Зрозумівши, наскільки точні прогнози, природно поставити запитання – які саме ознаки модель вважає визначальними. Відповідь дає аналіз важливості ознак.

Щоб навчену модель можна було використовувати поза середовищем Jupyter, три ключові об’єкти збережено у форматі pickle: саму модель (model.pkl), навчений масштабувальник (scaler.pkl) та перелік навчальних колонок у правильному порядку (model_columns.pkl). Останній файл є

критично важливим: він гарантує, що вектор ознак, сформований застосунком із введених користувачем даних, матиме точно таку саму структуру та порядок колонок, як і навчальна вибірка. Без цього прогноз був би некоректним.

```
model.pkl збережено
scaler.pkl збережено
model_columns.pkl збережено
Кількість ознак: 59
Тестовий прогноз: 10,320.30
Реальна ціна:     10,308.00
Всі файли збережено успішно!
```

Рисунок 3 .18 – Успішне збереження

Збережені артефакти стають основою для роботи кінцевого застосунку.

3.6 Вебзастосунок для прогнозування вартості

Практичним результатом роботи є вебзастосунок AutoPrice AI, побудований на фреймворку Streamlit. Саме він перетворює дослідницький прототип на готовий до використання інструмент і є головною практичною перевагою розробленої системи. Застосунок надає користувачеві зручний графічний інтерфейс, у якому достатньо вказати характеристики автомобіля, щоб миттєво отримати прогноз його ринкової вартості.

При запуску застосунок одноразово завантажує збережені артефакти моделі за допомогою механізму кешування, що забезпечує швидкий відгук інтерфейсу. Користувач задає параметри автомобіля через зручні елементи керування: марку та модель обирають зі списків, пов’язаних між собою (для кожної марки доступні лише її моделі), тип палива, коробку передач, кількість дверей та кількість попередніх власників – теж зі списків, а рік випуску, об’єм двигуна та пробіг – за допомогою повзунків. Загальний вигляд інтерфейсу подано на рисунку 3.19.

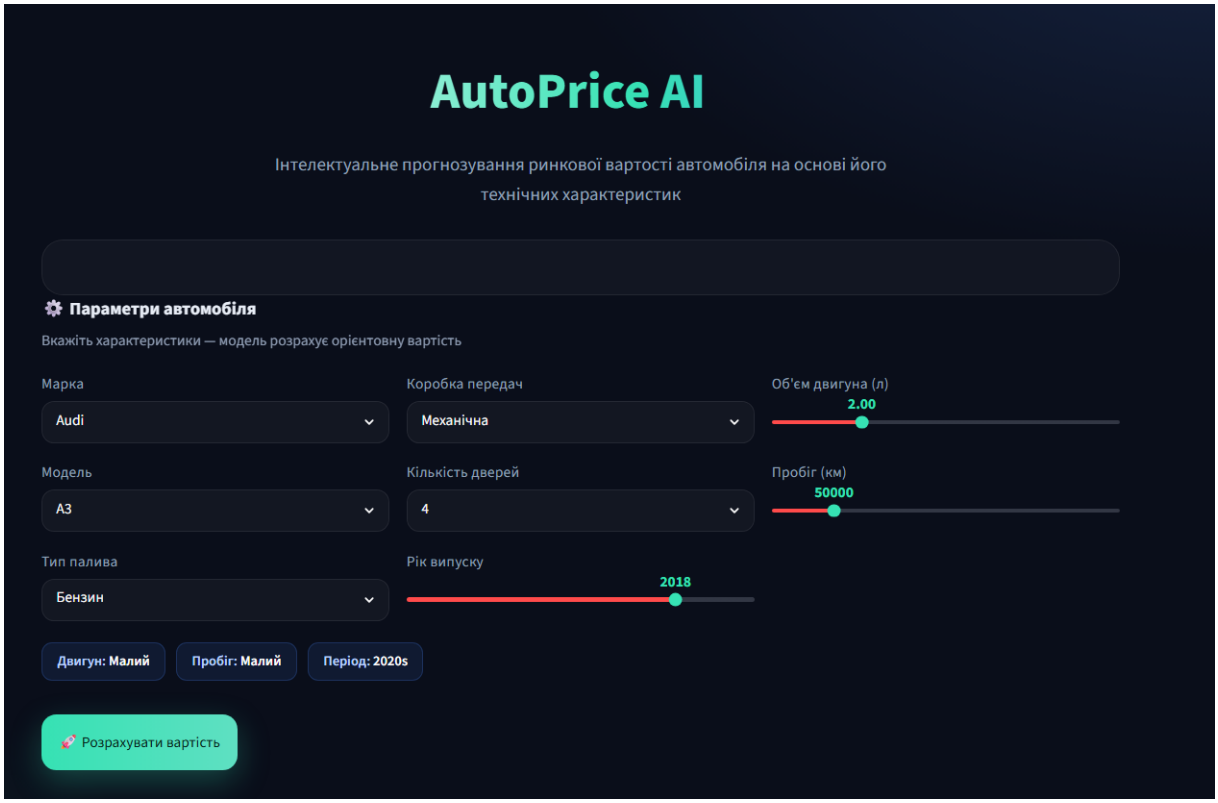

Рисунок 3.19 – Інтерфейс введення параметрів автомобіля

Інтерфейс охоплює повний набір ознак, на яких навчалася модель, включно з кількістю попередніх власників. Хоча розвідувальний аналіз показав слабкий вплив цього параметра на ціну в наявному наборі даних, його залишено в формі як природну та зрозумілу користувачеві характеристику; модель коректно обробляє будь–яке з п’яти можливих значень, де варіант «один власник» відповідає базовій категорії кодування.

Після натискання кнопки розрахунку застосунок відтворює ту саму послідовність перетворень, що й під час навчання: формує вектор числових ознак, масштабує його збереженим масштабувальником, додає One–Hot ознаки за вибраними категоріями (зокрема за кількістю власників) та вирівнює структуру вектора під навчальні колонки. Сформований вектор подається на вхід моделі, і отриманий прогноз виводиться користувачеві у вигляді ринкової вартості разом зі зведенням заданих характеристик. Приклад результату прогнозування показано на рисунку 3.20.

Рисунок 3.20 – Результат прогнозування вартості автомобіля

Узгодженість логіки застосунку з логікою навчання моделі гарантує, що прогнози у застосунку збігаються з тими, які дає модель у середовищі розробки, з точністю до кількох доларів. Таким чином, розроблений вебзастосунок повністю реалізує мету роботи – надає кінцевому користувачеві простий і точний інструмент оцінки ринкової вартості автомобіля на основі методів машинного навчання.

ВИСНОВКИ

У дипломній роботі розроблено інформаційну систему прогнозування ринкової вартості автомобілів на основі методів машинного навчання, яка охоплює предиктивне ядро для обробки даних та інтерактивний вебклієнт для взаємодії з кінцевим користувачем.

У першому розділі проаналізовано предметну область вторинного автомобільного ринку та розглянуто існуючі підходи й аналоги до оцінювання транспортних засобів. Аналіз показав, що більшість традиційних методів або спираються на суб'єктивну експертну оцінку, або використовують жорсткі статичні правила, які не здатні одночасно враховувати складні нелінійні взаємозв'язки між десятками технічних характеристик. На основі цього обґрунтовано необхідність розробки власної системи на базі алгоритмів штучного інтелекту, що здатна автоматично виявляти закономірності ціноутворення у великих масивах реальних ринкових даних.

У другому розділі виконано проєктування математичної та архітектурної складових системи. Розглянуто комплекс алгоритмів регресії – від базових лінійних моделей до сучасних ансамблевих фреймворків (Random Forest, CatBoost, XGBoost). Визначено систему метрик для об'єктивного оцінювання точності (MAE, MSE, RMSE, R^2). Спроєктовано пайплайн обробки даних, що передбачає робастне масштабування числових ознак та бінарне кодування категоріальних параметрів. Обґрунтовано клієнт– серверну взаємодію та архітектуру вебзастосунку з інкапсульованим предиктивним модулем.

У третьому розділі описано практичну реалізацію всіх підсистем в екосистемі мови Python. Проведено глибокий розвідувальний аналіз (EDA) набору даних із 10 000 записів, візуалізовано розподіли та виявлено ключові фактори впливу на ціну. Інженерію ознак реалізовано з використанням інструментів RobustScaler та One– Hot Encoding. Навчання моделей проведено із застосуванням крос– валідаційного пошуку оптимальних гіперпараметрів, де найвищу предиктивну точність продемонстрували ансамблеві алгоритми на

основі дерев рішень. Навчену математичну модель інтегровано у вебзастосунок на базі фреймворку Streamlit, що гарантує миттєвий відгук системи та зручне введення характеристик автомобіля через сучасний адаптивний інтерфейс.

Практичне значення розробленої системи полягає в тому, що вона забезпечує користувачу швидкий, об'єктивний та математично обґрунтований інструмент для визначення ринкової вартості транспортного засобу. Такий підхід усуває вплив людського фактору, нівелює ризики цінового шахрайства та є актуальним як для рядових покупців і продавців, так і для аналітиків автомобільного ринку.

Система є архітектурно цілісною і придатною до подальшого розширення – зокрема, додавання функціоналу для автоматичного парсингу свіжих оголошень з автомайданчиків для регулярного донавчання моделей, впровадження обробки природної мови для аналізу текстових описів, або ж інтеграції нейронних мереж комп'ютерного зору для оцінки стану автомобіля за фотографіями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скільки авто насправді в Україні та які можуть бути наслідки. РБК-Україна. URL: https://auto.rbc.ua/ukr/news/skilki-vsogo-ukrayini-mashin-chomu-mozhna-1750949039.html (дата звернення: 09.06.2026).
2. Що з цінами на вторинному авторинку? Підсумки січня 2025 року. Інститут досліджень авторинку. URL: https://eauto.org.ua/ (дата звернення: 09.06.2026).
3. Best Used Car Websites. Edmunds. URL: https://www.edmunds.com/car-buying/best-used-car-websites.html (дата звернення: 09.06.2026).
4. Регресійний аналіз. Wiki ТНТУ. URL: https://wiki.tntu.edu.ua/Регресійний_аналіз (дата звернення: 09.06.2026).
5. Машинне навчання: що це таке, як працює і для чого використовується. GoIT. URL: https://goit.global/ua/articles/mashynne-navchannia-shcho-tse-take-iak-pratsiuie-i-dlia-choho-vykorystovuietsia/ (дата звернення: 09.06.2026).
6. Машинне навчання простими словами. Частина 1. Механіко-математичний факультет Львівського національного університету імені Івана Франка. URL: http://mmf.lnu.edu.ua/ar/1739 (дата звернення: 09.06.2026).
7. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
8. Breiman L. Random Forests. *Machine Learning*. 2001. Vol. 45, No. 1. P. 5–32. DOI: 10.1023/A:1010933404324.
9. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, 2016. P. 785–794. DOI: 10.1145/2939672.2939785.

10. Ke G., Meng Q., Finley T. et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. Long Beach, 2017. P. 3146–3154.
11. Prokhorenkova L., Gusev G., Vorobev A., Dorogush A. V., Gulin A. CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*. Montréal, 2018. P. 6638–6648.
12. McKinney W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference (SciPy 2010)*. Austin, 2010. P. 56–61. DOI: 10.25080/Majora-92bf1922-00a.
13. Harris C. R., Millman K. J., van der Walt S. J. et al. Array programming with NumPy. *Nature*. 2020. Vol. 585, No. 7825. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
14. Hunter J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 2007. Vol. 9, No. 3. P. 90–95. DOI: 10.1109/MCSE.2007.55.
15. Waskom M. L. seaborn: statistical data visualization. *Journal of Open Source Software*. 2021. Vol. 6, No. 60. P. 3021. DOI: 10.21105/joss.03021.
16. Friedman J. H. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*. 2001. Vol. 29, No. 5. P. 1189–1232. DOI: 10.1214/aos/1013203451.
17. Linear Regression in Machine Learning. GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/ml-linear-regression/ (дата звернення: 09.06.2026).
18. Omarzai F. Decision Tree In Depth. Medium. 2024. URL: https://medium.com/@fraidoonomarzai99/decision-tree-in-depth-da6ff64d8e54 (дата звернення: 09.06.2026).
19. Random Forest Algorithm in Machine Learning. GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/ (дата звернення: 09.06.2026).

20. Car Prices Prediction Data. Kaggle. URL: https://www.kaggle.com/datasets/mrsimple07/car-prices-prediction-data (дата звернення: 09.06.2026).

ДОДАТОК А ЛІСТИНГ КОДУ НАВЧАННЯ МОДЕЛЕЙ

Лістинг програмного коду навчання моделей

А.1 Імпорт бібліотек та налаштування середовища

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import itertools
import plotly.graph_objects as go
import time

from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.exceptions import ConvergenceWarning
from sklearn.linear_model import LogisticRegression, LinearRegression, Ridge, Lasso,
ElasticNet
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split, cross_val_score,GridSearchCV
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, StandardScaler, RobustScaler

import warnings
warnings.simplefilter(action="ignore")

pd.set_option('display.max_columns', 500)
pd.set_option('display.width', None)
pd.set_option('display.max_rows', 20)
pd.set_option('display.float_format', lambda x: '%.3f' % x)
```

А.2 Завантаження набору даних та функція первинного аудиту

```
df = pd.read_csv('car_price_dataset.csv')

def check_df(dataframe, head=5):
    print("##################### Shape #####################")
    print(dataframe.shape)
    print("##################### Types #####################")
    print(dataframe.dtypes)
    print("##################### Head #####################")
    print(dataframe.head(head))
    print("##################### Tail #####################")
    print(dataframe.tail(head))
    print("##################### NA #####################")
    print(dataframe.isnull().sum())
    print("##################### Quantiles #####################")
```

```
    numeric_df = dataframe.select_dtypes(include="number")
    quantiles = numeric_df.quantile([0, 0.05, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99, 1]).T
    print(quantiles)

check_df(df)
```

A.3 Автоматичне визначення типів ознак

```
def grab_col_names(dataframe, cat_th=10, car_th=31):
    """
    Returns the names of categorical, numeric and categorical-but-cardinal variables.
    Works with pandas 3.0 where text columns have dtype 'str' (not object 'O').
    """
    # A column is "categorical-looking" if it is NOT numeric (text/str/object/category/bool)
    def is_cat_like(col):
        return not pd.api.types.is_numeric_dtype(dataframe[col])

    cat_cols = [col for col in dataframe.columns if is_cat_like(col)]

    num_but_cat = [col for col in dataframe.columns
                   if dataframe[col].nunique() < cat_th and
pd.api.types.is_numeric_dtype(dataframe[col])]

    cat_but_car = [col for col in dataframe.columns
                   if dataframe[col].nunique() > car_th and is_cat_like(col)]

    cat_cols = cat_cols + num_but_cat
    cat_cols = [col for col in cat_cols if col not in cat_but_car]

    num_cols = [col for col in dataframe.columns if
pd.api.types.is_numeric_dtype(dataframe[col])]
    num_cols = [col for col in num_cols if col not in num_but_cat]

    print(f"Observations: {dataframe.shape[0]}")
    print(f"Variables: {dataframe.shape[1]}")
    print(f'cat_cols: {len(cat_cols)}')
    print(f'num_cols: {len(num_cols)}')
    print(f'cat_but_car: {len(cat_but_car)}')
    print(f'num_but_cat: {len(num_but_cat)}')

    return cat_cols, num_cols, cat_but_car, num_but_cat

cat_cols, num_cols, cat_but_car,  num_but_cat = grab_col_names(df)
```

A.4 Аналіз категоріальних і числових ознак

```
def cat_summary(dataframe, col_name, plot=False):
    print(pd.DataFrame({col_name: dataframe[col_name].value_counts(),
                        "Ratio": 100 * dataframe[col_name].value_counts() / len(dataframe)}))
    print("##########################################")
    if plot:
        sns.countplot(x=dataframe[col_name], data=dataframe, palette="Set2")
        plt.xticks(rotation=45)
        plt.show(block=True)

for col in cat_cols:
```

```
    cat_summary(df, col, plot=True)

def num_summary(dataframe, col_name, plot=False):
    quantiles = [0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 0.95, 0.99]
    print(dataframe[col_name].describe(quantiles).T)

    if plot:
        sns.histplot(data=dataframe, x=col_name, bins=20, kde=True, color="blue")
        plt.xticks(rotation=45)
        plt.show(block=True)

for col in num_cols:
    num_summary(df, col, plot=True)
```

A.5 Аналіз залежності ціни від ознак

```
def target_summary_with_cat(dataframe, target, categorical_col, plot=False):
    print(pd.DataFrame({'TARGET_MEAN': dataframe.groupby(categorical_col)[target].mean()}),
end='\n\n\n')
    if plot:
        sns.barplot(x=categorical_col, y=target, data=dataframe, palette="Set2")
        plt.xticks(rotation=45)
        plt.show(block=True)

for col in cat_cols:
    target_summary_with_cat(df, "Price", col, plot=True)
```

A.6 Кореляційний аналіз та розподіл цільової змінної

```
def high_correlated_cols2(dataframe, plot=False, corr_th=0.70):
    # Sadece sayısal sütunları seçelim
    numeric_df = dataframe.select_dtypes(include=[np.number])

    # Korelasyon matrisi oluştur
    corr = numeric_df.corr()
    cor_matrix = corr.abs()

    # Üst üçgen matrisini seç (gereksiz tekrarları kaldırmak için)
    upper_triangle_matrix = cor_matrix.where(np.triu(np.ones(cor_matrix.shape),
k=1).astype(bool))

    # Korelasyonu eşikten yüksek olan sütun çiftlerini seçelim
    high_corrs = [(col, row)
                  for col in cor_matrix.columns
                  for row in cor_matrix.index
                  if not pd.isna(upper_triangle_matrix.loc[row, col]) and abs(
            upper_triangle_matrix.loc[row, col]) > corr_th]

    if plot:
        # Isı haritasını çiz
        plt.figure(figsize=(10, 6))
        sns.heatmap(corr, annot=True, fmt=".2f", cmap="RdBu")
        plt.title("Korelasyon Isı Haritası")
        plt.show()

    return high_corrs
```

```
high_correlated_cols2(df, plot=True)

df["Price"].hist(bins=100)
plt.show(block=True)

np.log1p(df['Price']).hist(bins=50)
plt.show(block=True)
```

A.7 Перевірка викидів і пропущених значень

```
def outlier_thresholds(dataframe, col_name, q1=0.25, q3=0.78):
    quartile1 = dataframe[col_name].quantile(q1)
    quartile3 = dataframe[col_name].quantile(q3)
    interquantile_range = quartile3 - quartile1
    up_limit = quartile3 + 1.5 * interquantile_range
    low_limit = quartile1 - 1.5 * interquantile_range
    return low_limit, up_limit

def check_outlier(dataframe, col_name):
    low_limit, up_limit = outlier_thresholds(dataframe, col_name)
    if dataframe[(dataframe[col_name] > up_limit) | (dataframe[col_name] <
low_limit)].any(axis=None):
        return True
    else:
        return False

for col in num_cols:
    print(col, check_outlier(df, col))

def missing_values_table(dataframe, na_name=False):
    na_columns = [col for col in dataframe.columns if dataframe[col].isnull().sum() > 0]

    n_miss = dataframe[na_columns].isnull().sum().sort_values(ascending=False)

    ratio = (dataframe[na_columns].isnull().sum() / dataframe.shape[0] *
100).sort_values(ascending=False)

    missing_df = pd.concat([n_miss, np.round(ratio, 2)], axis=1, keys=['n_miss', 'ratio'])

    print(missing_df, end="\n")

    if na_name:
        return na_columns

missing_values_table(df)
```

A.8 Конструювання ознак (групування) та One-Hot кодування

```
df["Engine_Size_Group"] = pd.cut(df["Engine_Size"], bins=3, labels=["Small", "Medium",
"Large"])
df["Mileage_Group"] = pd.cut(df["Mileage"], bins=3, labels=["Low", "Medium", "High"])
df["Year_of_Registration_Group"] = pd.cut(df["Year"], bins=3, labels=["2000s", "2010s",
"2020s"])

def one_hot_encoder(dataframe, categorical_cols, drop_first=False):
    dataframe = pd.get_dummies(dataframe, columns=categorical_cols, drop_first=drop_first)
```

```
    # get_dummies returns bool columns in modern pandas; cast to int for models/scaler
    bool_cols = dataframe.select_dtypes(include="bool").columns
    dataframe[bool_cols] = dataframe[bool_cols].astype(int)
    return dataframe

df = one_hot_encoder(df, cat_cols, drop_first=True)
```

A.9 Масштабування числових ознак та поділ на вибірки

```
num_cols = [col for col in num_cols if col not in ["Price"]]

scaler = RobustScaler()

df[num_cols] = scaler.fit_transform(df[num_cols])

y = df["Price"]

X = df.drop(["Price"], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=17)
```

A.10 Навчання та порівняння одинадцяти моделей

```
models = [("LR", LinearRegression()),
          ("Ridge", Ridge()),
          ("Lasso", Lasso()),
          ("ElasticNet", ElasticNet()),
          ("KNN", KNeighborsRegressor()),
          ("CART", DecisionTreeRegressor()),
          ("RF", RandomForestRegressor()),
          ("GBM", GradientBoostingRegressor()),
          ("XGBoost", XGBRegressor(objective="reg:squarederror")),
          ("LightGBM", LGBMRegressor(verbose=-1)),
          ("CatBoost", CatBoostRegressor(verbose=False))]

rmse_scores = []
r2_scores = []
mae_scores = []
mse_scores = []
execution_times = []

for name, regressor in models:
    start_time = time.time()
    regressor.fit(X_train, y_train)
    y_pred = regressor.predict(X_test)

    # RMSE через cross_val_score
    rmse = np.mean(np.sqrt(np.abs(cross_val_score(regressor, X, y, cv=5,
scoring="neg_mean_squared_error"))))
    rmse_scores.append(rmse)

    r2 = metrics.r2_score(y_test, y_pred)
    r2_scores.append(r2)

    mae = metrics.mean_absolute_error(y_test, y_pred)
    mae_scores.append(mae)
```

```
    mse = metrics.mean_squared_error(y_test, y_pred)
    mse_scores.append(mse)

    execution_time = time.time() - start_time
    execution_times.append(execution_time)

    print(f"RMSE: {round(rmse, 4)} ({name})")
    print(f"R^2 Score: {round(r2, 4)} ({name})")
    print(f"MAE: {round(mae, 4)} ({name})")
    print(f"MSE: {round(mse, 4)} ({name})")
    print(f"Execution Time: {round(execution_time, 2)} seconds\n")
```

A.11 Оптимізація гіперпараметрів (GridSearchCV) та вибір найкращої моделі

```
# Initialize the models
models = [('LR', LinearRegression()),
          ("Ridge", Ridge()),
          ("Lasso", Lasso()),
          ("ElasticNet", ElasticNet()),
          ('KNN', KNeighborsRegressor()),
          ('CART', DecisionTreeRegressor()),
          ('RF', RandomForestRegressor()),
          ('GBM', GradientBoostingRegressor()),
          ("XGBoost", XGBRegressor(objective='reg:squarederror')),
          ("LightGBM", LGBMRegressor(verbose=-1)),
          ("CatBoost", CatBoostRegressor(verbose=False))]

rmse_scores = []
r2_scores = []
mae_scores = []
mse_scores = []
execution_times = []

param_grids = {
    'LR': {},
    'Ridge': {'alpha': [0.1, 1.0]},
    'Lasso': {'alpha': [0.1, 1.0]},
    'ElasticNet': {'alpha': [0.1, 1.0], 'l1_ratio': [0.1, 0.9]},
    'KNN': {'n_neighbors': [3, 5]},
    'CART': {'max_depth': [None, 10], 'min_samples_leaf': [1, 2]},
    'RF': {'n_estimators': [10, 50], 'max_depth': [None, 10]},
    'GBM': {'n_estimators': [10, 50], 'learning_rate': [0.01, 0.1]},
    'XGBoost': {'n_estimators': [10, 50], 'learning_rate': [0.01, 0.1]},
    'LightGBM': {'n_estimators': [10, 50], 'learning_rate': [0.01, 0.1]},
    'CatBoost': {'iterations': [10, 50], 'learning_rate': [0.01, 0.1], 'depth': [4, 6]}
}

# Track the genuinely best model across all candidates (by R^2 on test set)
best_model = None
best_r2 = -np.inf
best_name = None

for name, regressor in models:
    print(f"Hyperparameter Tuning for {name}:")
    start_time = time.time()
```

```
    if param_grids[name]:
        grid_search = GridSearchCV(regressor, param_grid=param_grids[name], cv=5, n_jobs=-1)
        grid_search.fit(X_train, y_train)
        current_model = grid_search.best_estimator_
        print(f"Best parameters: {grid_search.best_params_}")
    else:
        current_model = regressor.fit(X_train, y_train)

    y_pred = current_model.predict(X_test)

    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    rmse_scores.append(rmse)
    r2 = r2_score(y_test, y_pred)
    r2_scores.append(r2)
    mae = mean_absolute_error(y_test, y_pred)
    mae_scores.append(mae)
    mse = mean_squared_error(y_test, y_pred)
    mse_scores.append(mse)
    execution_time = time.time() - start_time
    execution_times.append(execution_time)

    if r2 > best_r2:
        best_r2 = r2
        best_model = current_model
        best_name = name

    print(f"RMSE: {round(rmse, 4)} ({name})")
    print(f"R^2 Score: {round(r2, 4)} ({name})")
    print(f"MAE: {round(mae, 4)} ({name})")
    print(f"MSE: {round(mse, 4)} ({name})")
    print(f"Execution Time: {round(execution_time, 2)} seconds\n")

print(f"BEST MODEL: {best_name} (R^2 = {round(best_r2, 4)})")

# Plot RMSE scores
plt.figure(figsize=(10, 6))
plt.bar([name for name, _ in models], rmse_scores)
plt.xticks(rotation=45)
plt.xlabel("Model"); plt.ylabel("RMSE"); plt.title("Model Performance (RMSE)")
plt.tight_layout(); plt.show()

# Plot R^2 scores
plt.figure(figsize=(10, 6))
plt.bar([name for name, _ in models], r2_scores)
plt.xticks(rotation=45)
plt.xlabel("Model"); plt.ylabel("R^2 Score"); plt.title("Model Performance (R^2 Score)")
plt.tight_layout(); plt.show()
```

A.12 Аналіз точності прогнозів та важливості ознак

```
# Final Prediction Model
final_model = best_model

# Make predictions on the test set using the final model
y_final_pred = final_model.predict(X_test)
final_y_pred = (y_final_pred)
final_y_test =(y_test)

# Create a DataFrame with the predicted prices and true prices
```

```
results = pd.DataFrame({'Predicted Price': final_y_pred, 'True Price': final_y_test})

# Calculate the difference between the true prices and predicted prices and add a new column
results['Difference'] = results['True Price'] - results['Predicted Price']

# Display the results
print(results)

def plot_importance(model, features, num=50, save=False):
    if not hasattr(model, "feature_importances_"):
        print(f"Модель {type(model).__name__} не підтримує feature_importances_.
Пропускаємо.")
        return
    feature_imp = pd.DataFrame({"Value": model.feature_importances_, "Feature":
features.columns})
    plt.figure(figsize=(10, 10))
    sns.set(font_scale=1)
    sns.barplot(x="Value", y="Feature", palette="Set2",
                data=feature_imp.sort_values(by="Value", ascending=False)[0:num])
    plt.title("Features")
    plt.tight_layout()
    plt.show(block=True)
    if save:
        plt.savefig("importances.png")

plot_importance(final_model, X)
```

A.13 Збереження артефактів моделі

```
import pickle as pk

# Зберігаємо найкращу модель
pk.dump(best_model, open("model.pkl", "wb"))
print("model.pkl збережено")

# Зберігаємо scaler
pk.dump(scaler, open("scaler.pkl", "wb"))
print("scaler.pkl збережено")

# Зберігаємо список колонок X
pk.dump(list(X.columns), open("model_columns.pkl", "wb"))
print("model_columns.pkl збережено")
print(f"Кількість ознак: {len(X.columns)}")

# Перевірка
test_pred = best_model.predict(X_test[:1])
print(f"Тестовий прогноз: {test_pred[0]:,.2f}")
print(f"Реальна ціна:     {y_test.iloc[0]:,.2f}")
print("Всі файли збережено успішно!")
```

ДОДАТОК Б ЛІСТИНГ КОДУ ВЕБЗАСТОСУНКУ

Вміст файлу app.py

```
import pandas as pd

import numpy as np

import pickle as pk

import streamlit as st


# 1. НАЛАШТУВАННЯ СТОРІНКИ ТА МОДЕЛІ

st.set_page_config(

    page_title="AutoPrice AI",

    page_icon="🚗",

    layout="wide",

    initial_sidebar_state="collapsed",

)


# ---------- Сучасний кастомний стиль ----------

st.markdown("""

<style>

@import
url('https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700;800&display=swap'
);


:root {

    --bg-0: #0a0e1a;

    --bg-1: #111726;

    --bg-2: #18203200;

    --card: rgba(255,255,255,0.035);

    --card-border: rgba(255,255,255,0.08);

    --accent: #36e2b4;

    --accent-2: #4f8cff;

    --text: #e9eef7;

    --text-dim: #93a1b8;

}


.stApp {

    background:

        radial-gradient(1100px 600px at 85% -10%, rgba(79,140,255,0.18), transparent 60%),
```

```
        radial-gradient(900px 500px at 0% 110%, rgba(54,226,180,0.14), transparent 55%),
        var(--bg-0);
    color: var(--text);
    font-family: 'Inter', sans-serif;
}

/* Сховати дефолтні елементи Streamlit */
#MainMenu, footer, header {visibility: hidden;}
.block-container {padding-top: 2.2rem; padding-bottom: 3rem; max-width: 1180px;}

/* ---------- Hero ---------- */
.hero {
    text-align: center;
    padding: 1.2rem 0 0.4rem;
}
.hero h1 {
    font-size: 3.0rem; font-weight: 800; line-height: 1.05; margin: 0;
    background: linear-gradient(90deg, #ffffff 10%, var(--accent) 55%, var(--accent-2) 100%);
    -webkit-background-clip: text; -webkit-text-fill-color: transparent;
    background-clip: text;
}
.hero p {
    color: var(--text-dim); font-size: 1.08rem; font-weight: 400;
    max-width: 620px; margin: 0.9rem auto 0;
}

/* ---------- Картки-секції ---------- */
.section-card {
    background: var(--card);
    border: 1px solid var(--card-border);
    border-radius: 20px;
    padding: 1.6rem 1.7rem;
    backdrop-filter: blur(12px);
    margin-top: 1.4rem;
}
.section-title {
    font-size: 1.05rem; font-weight: 700; color: var(--text);
    display: flex; align-items: center; gap: 10px; margin-bottom: 0.3rem;
```

```
}
.section-sub { color: var(--text-dim); font-size: 0.88rem; margin-bottom: 1.1rem;}

/* ---------- Поля вводу ---------- */
.stSelectbox label, .stSlider label {
    color: var(--text-dim) !important; font-weight: 600 !important;
    font-size: 0.82rem !important; letter-spacing: 0.02em;
}
div[data-baseweb="select"] > div {
    background: rgba(255,255,255,0.04) !important;
    border: 1px solid var(--card-border) !important;
    border-radius: 12px !important;
}
div[data-baseweb="select"] > div:hover { border-color: var(--accent) !important; }

/* Слайдери */
.stSlider [data-baseweb="slider"] [role="slider"] { background: var(--accent) !important; }
div[data-testid="stSliderThumbValue"] { color: var(--accent) !important; font-weight: 700; }

/* Чіпи-теги груп */
.chips { display: flex; gap: 10px; flex-wrap: wrap; margin-top: 4px;}
.chip {
    padding: 7px 14px; border-radius: 10px; font-size: 0.82rem; font-weight: 600;
    background: rgba(79,140,255,0.10); border: 1px solid rgba(79,140,255,0.22);
    color: #bcd2ff;
}
.chip b { color: #fff; }

/* ---------- Кнопка ---------- */
.stButton > button {
    width: 100%; border: none; border-radius: 14px;
    padding: 0.95rem 1rem; font-size: 1.02rem; font-weight: 700;
    color: #06231b;
    background: linear-gradient(90deg, var(--accent), #5fe0c2);
    box-shadow: 0 10px 30px rgba(54,226,180,0.28);
    transition: transform .15s ease, box-shadow .15s ease;
}
.stButton > button:hover {
```

```
    transform: translateY(-2px);
    box-shadow: 0 14px 38px rgba(54,226,180,0.4);
    color: #06231b;
}

/* ---------- Результат ---------- */
.result {
    margin-top: 1.5rem; text-align: center;
    background: linear-gradient(135deg, rgba(54,226,180,0.12), rgba(79,140,255,0.10));
    border: 1px solid rgba(54,226,180,0.30);
    border-radius: 22px; padding: 2rem 1.5rem;
}
.result-label { color: var(--text-dim); font-size: 0.9rem; font-weight: 600;
    text-transform: uppercase; letter-spacing: 0.08em; }
.result-value {
    font-size: 3.2rem; font-weight: 800; margin: 0.3rem 0 0;
    background: linear-gradient(90deg, var(--accent), var(--accent-2));
    -webkit-background-clip: text; -webkit-text-fill-color: transparent;
}
.result-grid {
    display: grid; grid-template-columns: repeat(auto-fit, minmax(150px,1fr));
    gap: 12px; margin-top: 1.5rem;
}
.spec {
    background: rgba(255,255,255,0.04); border: 1px solid var(--card-border);
    border-radius: 12px; padding: 0.8rem 1rem; text-align: left;
}
.spec .k { color: var(--text-dim); font-size: 0.72rem; text-transform: uppercase;
    letter-spacing: 0.05em;}
.spec .v { color: var(--text); font-size: 1.0rem; font-weight: 700; margin-top: 2px;}
</style>
""", unsafe_allow_html=True)

# ---------- Завантаження артефактів моделі ----------
@st.cache_resource
def load_artifacts():
    model = pk.load(open('model.pkl', 'rb'))
    scaler = pk.load(open('scaler.pkl', 'rb'))
```

```
    model_columns = pk.load(open('model_columns.pkl', 'rb'))
    return model, scaler, model_columns

try:
    model, scaler, model_columns = load_artifacts()
except FileNotFoundError:
    st.error("Файли моделі (model.pkl / scaler.pkl / model_columns.pkl) не знайдено! "
             "Спочатку запустіть Jupyter Notebook для їх генерації.")
    st.stop()

# 2. HERO
st.markdown("""
<div class="hero">
    <h1>AutoPrice AI</h1>
    <p>Інтелектуальне прогнозування ринкової вартості автомобіля
       на основі його технічних характеристик</p>
</div>
""", unsafe_allow_html=True)

# 3. ДАНІ ДЛЯ ІНТЕРФЕЙСУ
brand_models = {
    'Audi':       ['A3', 'A4', 'Q5'],
    'BMW':        ['3 Series', '5 Series', 'X5'],
    'Chevrolet':  ['Equinox', 'Impala', 'Malibu'],
    'Ford':       ['Explorer', 'Fiesta', 'Focus'],
    'Honda':      ['Accord', 'CR-V', 'Civic'],
    'Hyundai':    ['Elantra', 'Sonata', 'Tucson'],
    'Kia':        ['Optima', 'Rio', 'Sportage'],
    'Mercedes':   ['C-Class', 'E-Class', 'GLA'],
    'Toyota':     ['Camry', 'Corolla', 'RAV4'],
    'Volkswagen': ['Golf', 'Passat', 'Tiguan'],
}
brands        = sorted(brand_models.keys())

# Значення для моделі (англійські технічні ключі) та їх відображення українською.
# У списках користувач бачить українські підписи, а в модель передаються коректні
# англомовні категорії, на яких вона навчалася.
fuel_ua = {
```

```
    'Petrol':   'Бензин',
    'Diesel':   'Дизель',
    'Electric': 'Електро',
    'Hybrid':   'Гібрид',
}
trans_ua = {
    'Manual':         'Механічна',
    'Automatic':      'Автоматична',
    'Semi-Automatic': 'Напівавтоматична',
}
# Зворотні мапи: український підпис -> технічний ключ
fuel_ua_inv  = {v: k for k, v in fuel_ua.items()}
trans_ua_inv = {v: k for k, v in trans_ua.items()}

# Переклад груп (Feature Engineering) для відображення в чіпах
engine_group_ua  = {'Small': 'Малий', 'Medium': 'Середній', 'Large': 'Великий'}
mileage_group_ua = {'Low': 'Малий', 'Medium': 'Середній', 'High': 'Великий'}

fuel_types    = list(fuel_ua.values())
transmissions = list(trans_ua.values())

ENGINE_EDGES = (2.333, 3.667)
MILEAGE_EDGES = (99999.0, 199973.0)
YEAR_EDGES = (2007.667, 2015.333)

# 4. ФОРМА ВВОДУ
st.markdown('<div class="section-card">', unsafe_allow_html=True)
st.markdown("""
<div class="section-title">⚙️ Параметри автомобіля</div>
<div class="section-sub">Вкажіть характеристики — модель розрахує орієнтовну вартість</div>
""", unsafe_allow_html=True)

col1, col2, col3 = st.columns(3)

with col1:
    brand          = st.selectbox('Марка', brands)
    model_name     = st.selectbox('Модель', brand_models[brand])
    fuel_type_ua   = st.selectbox('Тип палива', fuel_types)
```

```
with col2:
    transmission_ua = st.selectbox('Коробка передач', transmissions)
    doors           = st.selectbox('Кількість дверей', [2, 3, 4, 5], index=2)
    year            = st.slider('Рік випуску', 2000, 2023, 2018)

with col3:
    engine_size  = st.slider("Об'єм двигуна (л)", 1.0, 5.0, 2.0, step=0.1)
    mileage      = st.slider('Пробіг (км)', 0, 300000, 50000, step=1000)

# Технічні (англомовні) ключі для моделі
fuel_type    = fuel_ua_inv[fuel_type_ua]
transmission = trans_ua_inv[transmission_ua]

# Owner_Count не виноситься в інтерфейс: у наборі даних ця ознака не корелює з ціною
# (кореляція ≈ 0). Модель навчалася з drop_first=True, де Owner_Count=1 — базова
# категорія, тож фіксуємо значення 1 (reindex усе одно обнуляє відповідні колонки).
owner_count = 1

# Групи (Feature Engineering як у ноутбуці)
engine_label = ('Small' if engine_size < ENGINE_EDGES[0]
                else ('Medium' if engine_size < ENGINE_EDGES[1] else 'Large'))
mileage_label = ('Low' if mileage < MILEAGE_EDGES[0]
                 else ('Medium' if mileage < MILEAGE_EDGES[1] else 'High'))
year_label = ('2000s' if year < YEAR_EDGES[0]
              else ('2010s' if year < YEAR_EDGES[1] else '2020s'))

st.markdown(f"""
<div class="chips">
    <div class="chip">Двигун: <b>{engine_group_ua[engine_label]}</b></div>
    <div class="chip">Пробіг: <b>{mileage_group_ua[mileage_label]}</b></div>
    <div class="chip">Період: <b>{year_label}</b></div>
</div>
""", unsafe_allow_html=True)

st.markdown('</div>', unsafe_allow_html=True)

st.write("")
```

```
predict = st.button("🚀 Розрахувати вартість")


# 5. ПРОГНОЗУВАННЯ
if predict:
    # Числові ознаки (масштабуються лише ці три)
    input_df = pd.DataFrame([{
        'Year':        float(year),
        'Engine_Size': float(engine_size),
        'Mileage':     float(mileage),
    }])
    scale_cols = ['Year', 'Engine_Size', 'Mileage']
    input_df[scale_cols] = scaler.transform(input_df[scale_cols])

    # One-Hot (drop_first=True; reindex вирівняє під навчальні колонки)
    input_df[f'Brand_{brand}'] = 1
    input_df[f'Model_{model_name}'] = 1
    input_df[f'Fuel_Type_{fuel_type}'] = 1
    input_df[f'Transmission_{transmission}'] = 1
    input_df[f'Doors_{doors}'] = 1
    input_df[f'Owner_Count_{owner_count}'] = 1
    input_df[f'Engine_Size_Group_{engine_label}'] = 1
    input_df[f'Mileage_Group_{mileage_label}'] = 1
    input_df[f'Year_of_Registration_Group_{year_label}'] = 1

    input_df = input_df.reindex(columns=model_columns, fill_value=0)

    car_price = float(model.predict(input_df)[0])

    st.markdown(f"""
    <div class="result">
        <div class="result-label">Орієнтовна ринкова вартість</div>
        <div class="result-value">${car_price:,.0f}</div>
        <div class="result-grid">
            <div class="spec"><div class="k">Авто</div><div class="v">{brand}
{model_name}</div></div>
            <div class="spec"><div class="k">Рік</div><div class="v">{year}</div></div>
            <div class="spec"><div class="k">Двигун</div><div class="v">{engine_size}
л</div></div>
```

```
            <div class="spec"><div class="k">Паливо</div><div
class="v">{fuel_type_ua}</div></div>

            <div class="spec"><div class="k">КПП</div><div
class="v">{transmission_ua}</div></div>

            <div class="spec"><div class="k">Пробіг</div><div class="v">{mileage:,}
км</div></div>

            <div class="spec"><div class="k">Дверей</div><div class="v">{doors}</div></div>

        </div>

    </div>

    """, unsafe_allow_html=True)
```

Бібліографічна довідка

Тема роботи: Розроблення інформаційної системи прогнозування ціни автомобілів на основі їх характеристик з використанням методів машинного навчання.

Обсяг бакалаврської роботи 75 сторінок.

Перелік графічного матеріалу:

1. КРБ.ІСТ - 017.00.00.001. Програмні вікна інформаційної системи прогнозування ціни.
2. КРБ.ІСТ - 017.00.00.002. Аналіз лінійних взаємозв'язків між технічними характеристиками та вартістю автомобілів.
3. КРБ.ІСТ - 017.00.00.003. Графічна візуалізація вхідного набору даних

Дата закінчення БР

Студент Микола СЛЮСАРЧУК