МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО

ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **“Прогнозування ринкової ціни вживаних автомобілів із використанням алгоритмів машинного навчання”**

Виконав: студент гр. Іт-61

Спеціальності 126 «Інформаційні системи та технології»
(шифр і назва)

Тимоць Ігор Богданович
(Прізвище та ініціали)

Керівник: к.т.н., доц. Лиса О.В.
(Прізвище та ініціали)

Рецензенти: д.т.н., проф. Власовець В.М.
(Прізвище та ініціали)

ЛЬВІВ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

"ЗАТВЕРДЖУЮ"
Завідувач кафедри_____
д.т.н., проф. А.М. Тригуба
"____"_____2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Тимоць Ігор Богданович

1. Тема роботи: «Прогнозування ринкової ціни вживаних автомобілів із використанням алгоритмів машинного навчання»

Керівник роботи Лиса Ольга Володимирівна, к.т.н., доцент.
Затверджені наказом по університету від 28.02 2025 року № 140 /к-с.

2. Строк подання студентом роботи 05.12.2025 р.

3. Вихідні дані до роботи: *Онлайн-платформи оголошень з джерелами реальних даних про ринок вживаних автомобілів, програмне середовище Python із бібліотеками NumPy, Pandas, Scikit-learn, XGBoost та Matplotlib, алгоритми машинного навчання*

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

Вступ.
1. Теоретичні основи прогнозування ринкових цін.
2. Розробка та навчання моделей машинного навчання.
3. Впровадження та апробація розробленої системи.
4. Охорона праці та безпека у надзвичайних ситуаціях.
5. Визначення ефективності від впровадження інформаційної системи прогнозування ринкової ціни вживаних автомобілів.
Висновки та пропозиції.
Список використаної літератури

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): *Типовий набір ознак для визначення вартості вживаних автомобілів; Основні характеристики даних; Схема очищення та препроцесинг даних; Схема процесу вибору та кодування ознак; Процес візуального та кореляційного аналізу даних; Архітектурна схема; Інтерфейс; Порівняння прогнозної ціни з ринковим діапазоном; Порівняння реальних та прогнозованих значень;* Порівняння реальних і прогнозних цін на автомобілі; *Графік важливості ознак; Розрахунок економічної ефективності.*

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	*Лиса О.В., доцент кафедри інформаційних технологій*		
4	*Городецький І.М., доцент кафедри інженерної механіки*		

7. Дата видачі завдання 1 березня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Приміт-ка
1	*Написання першого розділу*	01.03.25-04.05.25	
2	*Виконання другого розділу та аркушів ілюстраційного матеріалу до нього*	05.05.25-14.07.25	
3.	*Виконання третього розділу та аркушів ілюстраційного матеріалу до нього*	15.07.25-24.09.25	
4.	*Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»*	25.09.25-10.10.25	
5.	*Оцінення ефективності запропонованої системи*	11.10.25-31.10.25	
6.	*Завершення оформлення розрахунково-пояснювальної записки та презентації*	01.11.25-30.11.25	
7.	*Завершення роботи в цілому*	01.12.25-05.12.25	

Студент _____ Тимоць І.Б.
(підпис)

Керівник роботи _____ Лиса О.В.
(підпис)

УДК 621.311.1:004.8

Прогнозування ринкової ціни вживаних автомобілів із використанням алгоритмів машинного навчання. Тимоць І.Б. Кафедра інформаційних технологій – Львів, ЛНУВМБ ім. С.З.Ґжицького, 2025. Кваліфікаційна робота: 76 с. текст. част., 8 рис., 5 табл., 10 арк. ілюстраційного матеріалу, 40 джерел.

У роботі розглянуто застосування алгоритмів машинного навчання для прогнозування ринкової ціни вживаних автомобілів на основі їх технічних та експлуатаційних характеристик. У вступі обґрунтовано актуальність теми, сформульовано мету, завдання та об’єкт і предмет дослідження.

У першому розділі проаналізовано сучасний стан ринку вживаних автомобілів та існуючі підходи до прогнозування цін, розглянуто основні алгоритми машинного навчання, придатні для задач регресії, здійснено формалізацію задачі прогнозування та визначено вимоги до точності й ефективності моделей. Другий розділ присвячено процесу розробки та навчання моделей: описано методи збору й підготовки даних, підходи до поділу вибірки на навчальну, валідаційну та тестову, проведено візуальний і кореляційний аналіз, обґрунтовано вибір алгоритмів, налаштовано гіперпараметри та виконано навчання моделей із подальшим вибором оптимальної. У третьому розділі описано архітектуру програмного рішення, реалізацію користувацького інтерфейсу, інтеграцію моделі прогнозування у програмний продукт, а також результати тестування системи на реальних даних і аналіз ефективності її роботи.

У четвертому розділі розглянуто питання охорони праці та безпеки у надзвичайних ситуаціях. У п’ятому розділі наведено оцінку економічної ефективності впровадження інформаційної системи прогнозування ринкової ціни вживаних автомобілів та обґрунтовано доцільність її практичного використання.

Ключові слова: машинне навчання, прогнозування, ринкова ціна, вживані автомобілі, регресійні моделі, інформаційні системи, аналіз даних, штучний інтелект.

ЗМІСТ

ВСТУП

Ринок вживаних автомобілів є однією з найбільш динамічних та конкурентних сфер економіки, що постійно змінюється під впливом різноманітних факторів. Ціна транспортного засобу залежить не лише від його технічних параметрів (марка, модель, рік випуску, пробіг, об’єм двигуна, тип палива тощо), а й від суб’єктивних чинників, таких як попит і пропозиція на регіональному ринку, сезонність продажів, економічна ситуація в країні, а також навіть соціальні тренди. Традиційні методи оцінки, що ґрунтуються на досвіді експертів чи статистичних моделях, часто не забезпечують належної об’єктивності та точності, оскільки не враховують складних взаємозв’язків між багатьма факторами.

Зростання обсягів доступних даних завдяки розвитку електронних торговельних майданчиків, онлайн-платформ з продажу автомобілів та спеціалізованих баз даних створює передумови для впровадження сучасних підходів до аналізу. У цьому контексті технології штучного інтелекту, зокрема алгоритми машинного навчання, стають потужним інструментом, здатним виявляти приховані закономірності та будувати більш точні прогнози ринкової вартості транспортних засобів.

Використання методів машинного навчання дає можливість автоматизувати процес оцінювання вартості автомобілів; зменшити суб’єктивний вплив експертних оцінок; підвищити точність прогнозів завдяки аналізу багатофакторних залежностей; адаптувати моделі до змін ринкових умов у реальному часі. Практична значущість проблеми зумовлена тим, що наявність точних і зручних у використанні інструментів прогнозування ціни вживаних автомобілів є корисною як для покупців і продавців, так і для страхових компаній, банківських установ (у сфері кредитування та лізингу), дилерських мереж та онлайн-платформ продажу. Для споживачів це означає можливість уникнути переплати або продажу автомобіля за заниженою ціною, а для компаній — підвищення ефективності бізнес-процесів і зниження ризиків.

З наукової точки зору, актуальність дослідження обумовлена потребою у порівняльному аналізі різних алгоритмів машинного навчання та визначенні найбільш ефективних підходів для задачі прогнозування цін на основі реальних ринкових даних. Це відкриває перспективи створення нових інформаційних систем та сервісів, які можуть масштабуватися і застосовуватися на глобальному рівні. Отже, дослідження є актуальним як у науково-методичному, так і в практичному аспекті, оскільки воно спрямоване на поєднання сучасних інтелектуальних технологій з реальною економічною задачею, що має високу прикладну цінність.

Мета роботи – розробка та дослідження моделей машинного навчання для прогнозування ринкової вартості вживаних автомобілів на основі їх технічних і експлуатаційних характеристик.

Об’єкт дослідження – процес прогнозування ринкової вартості вживаних автомобілів.

Предмет дослідження – алгоритми машинного навчання та їх застосування для моделювання залежності вартості автомобіля від його характеристик. Наукова новизна полягає у комплексному застосуванні сучасних алгоритмів машинного навчання для вирішення задачі прогнозування ринкової вартості вживаних автомобілів із використанням багатофакторних даних.

РОЗДІЛ 1.
ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ РИНКОВИХ ЦІН

1.1. Аналіз ринку вживаних автомобілів та методи прогнозування ринкових цін

Ринок вживаних автомобілів є важливим сегментом світової та національної економіки. За даними міжнародних аналітичних агентств, у більшості країн світу обсяги продажу автомобілів, що вже перебували в експлуатації, перевищують продажі нових транспортних засобів. Це зумовлено більш доступною вартістю, широким вибором моделей та високим попитом серед населення із середнім рівнем доходу.

Особливістю цього ринку є його висока варіативність - ціна автомобіля формується під впливом не лише технічних характеристик, а й факторів макроекономічного та соціального характеру, зокрема високий рівень сегментації, що визначається маркою, моделлю, роком випуску, пробігом, технічним станом, комплектацією; непрозорість ціноутворення, оскільки вартість одного і того ж автомобіля може істотно відрізнятися залежно від регіону, продавця та умов продажу; швидка амортизація вартості, коли ціна нового автомобіля може знизитися на 20–30% вже в перший рік експлуатації; залежність від макроекономічних факторів (валютні коливання, зміни податкового законодавства, імпортні мита, вартість пального).

Серед ключових тенденцій, що визначають розвиток ринку вживаних автомобілів у світі та в Україні, можна виділити: зростання частки онлайн-продажів та електронних платформ (наприклад, AutoScout24, Copart, Auto.ria, OLX); підвищення значущості прозорості угод і довіри до інформації про технічний стан транспортного засобу; вплив глобальних економічних факторів (інфляція, валютні коливання, вартість пального) на ціноутворення; збільшення частки електромобілів та гібридів у структурі пропозицій; необхідність швидкої та об’єктивної оцінки вартості автомобіля як для покупців, так і для продавців.

Отже, ринок вживаних автомобілів вимагає сучасних інструментів аналітики, що дозволяють точно й оперативно прогнозувати справедливу ринкову ціну транспортного засобу.

Традиційно оцінка ринкової вартості автомобіля ґрунтувалася на таких підходах: експертні методи, коли ціну визначає фахівець на основі власного досвіду; порівняльний аналіз, що передбачає зіставлення транспортного засобу з аналогами, виставленими на ринку; статистичні методи (лінійна регресія, аналіз середніх цін), які враховують залежність вартості від окремих ключових характеристик.

Лінійна регресія є одним із базових статистичних інструментів прогнозування. Вона використовується для встановлення залежності ринкової ціни автомобіля від його характеристик (пробіг, рік випуску, марка, об'єм двигуна, тип палива тощо). Модель має вигляд:

$$Y=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n+\varepsilon, \quad (1.1)$$

де Y – ринкова ціна автомобіля, X_i – фактори (характеристики),

β_i – параметри моделі, ε – випадкова похибка.

Коефіцієнт β_i показує, як змінюється ціна авто при зміні певної характеристики на одиницю, за умови фіксованості інших змінних. Наприклад: кожні додаткові 10 000 км пробігу зменшують середню ринкову вартість автомобіля на певний відсоток (наприклад, 3–5%). Переваги - простота використання та інтерпретації; можливість оцінки впливу окремих факторів; базовий орієнтир для складніших моделей. Недоліки - погано працює з нелінійними залежностями; чутлива до мультиколінеарності факторів (наприклад, «вік авто» і «пробіг» часто сильно корелюють).

Аналіз середніх цін (hedonic pricing / порівняльний аналіз) використовується для оцінки ринкової вартості шляхом узагальнення інформації про ціни аналогічних автомобілів. Сутність методу: ринок розбивається на групи за ключовими характеристиками (марка, модель, рік випуску, пробіг, об'єм двигуна); для кожної групи обчислюється середня або медіанна ціна; прогнозна вартість нового об'єкта визначається як середнє значення для «подібних авто».

Формула для середньої арифметичної:

$$\overline{P} = \frac{1}{n} \sum_{i=1}^{n} P_i, \tag{1.2}$$

де $\overline{P}$ – середня ринкова ціна для групи,

P_i – ціна і-го автомобіля, n – кількість авто у групі.

Варіації методу - медіанна ціна (менш чутлива до викидів); зважене середнє (вага враховує додаткові фактори – технічний стан, комплектацію). Переваги - простота реалізації; зрозумілість для бізнес-користувачів і покупців. Недоліки - не враховує складної взаємодії характеристик; потребує великої кількості даних для достовірності; результати чутливі до наявності «аномальних» цін.

Лінійна регресія дозволяє формалізувати залежність ціни від характеристик і оцінити внесок кожної змінної. Аналіз середніх цін простіший, але менш точний; застосовується як базовий підхід або для валідації результатів моделей. Таким чином, обидва методи відіграють важливу роль у статистичному аналізі та є відправною точкою перед використанням сучасних алгоритмів машинного навчання. Сучасні методи прогнозування ринкових цін автомобілів базуються на математичному моделюванні, алгоритмах машинного навчання (ML) та штучного інтелекту (AI). Вони дозволяють аналізувати великі масиви даних, виявляти приховані закономірності, враховувати складні взаємозв’язки між характеристиками авто та підвищувати точність прогнозів у порівнянні з традиційними статистичними методами. Найбільш поширені підходи:

1. Розширений регресійний аналіз. Множинна регресія дозволяє враховувати одночасно кілька факторів, що впливають на ціну (пробіг, рік випуску, марка, технічний стан). Поліноміальна регресія застосовується, коли залежність ціни від характеристик є нелінійною. Їх перевагою є інтерпретованість результатів та можливість оцінки внеску кожної змінної у ціну. Проте складно враховувати взаємодію великої кількості факторів без ризику мультиколінеарності.

2. Ансамблеві методи - Random Forest, Gradient Boosting, XGBoost, LightGBM. Ідея полягає у поєднанні кількох моделей (дерев рішень), що підвищує

стабільність і точність прогнозу. Використання ансамблів дозволяє моделі краще справлятися з великими і складними даними, виявляти нелінійні взаємозв'язки та зменшувати ризик перенавчання.

3. Методи глибинного навчання (Deep Learning) - використовуються штучні нейронні мережі (ANN, DNN) для прогнозування цін на основі великої кількості вхідних змінних. Мережі здатні виявляти складні нелінійні залежності та взаємозв'язки між ознаками, що неможливо реалізувати простими регресійними моделями. Підходи глибинного навчання особливо ефективні при наявності великих обсягів даних та високої розмірності ознак.

4. Гібридні моделі поєднують кілька методів одночасно, наприклад, регресію + ансамблі, або ансамблі + нейронні мережі. Мета: досягнення максимальної точності прогнозу та зниження похибки. Ці моделі особливо корисні, коли дані мають різні типи ознак (категоріальні, числові, текстові) або коли окремі алгоритми показують неоднорідну продуктивність.

Таблиця 1.1

Порівняння традиційних та сучасних методів прогнозування ринкових цін на автомобілі

Категорія методів	Конкретні методи	Переваги	Недоліки	Особливості застосування
Традиційні статистичні	Лінійна регресія, аналіз середніх цін, гедонічне ціноутворення	Простота, зрозумілість, інтерпретація коефіцієнтів, базові оцінки залежностей	Погано працюють з нелінійними зв'язками, обмежені можливості для великих даних	Добре підходять для початкового аналізу та валідації моделей; оцінка впливу окремих факторів
Сучасні ML/AI	Множинна/поліноміальна регресія, ансамблеві методи (Random Forest, Gradient Boosting, XGBoost), нейронні мережі, гібридні моделі	Висока точність, здатність обробляти великі дані, виявляти складні залежності, знижувати похибку прогнозу	Складність інтерпретації, потребують великих наборів даних, висока обчислювальна вартість	Використовуються для автоматизованого прогнозування, складних сценаріїв ціноутворення, оптимізації моделей продажу та оцінки ризиків
Гібридні моделі	Комбінації ансамблів + нейронні мережі або регресія + бустинг	Поєднують переваги різних підходів, підвищують стабільність та точність	Високі вимоги до обчислень та даних, складність налаштування	Оптимальні для прогнозування цін на великі та неоднорідні набори даних, де окремі методи показують обмежену ефективність

Сучасні підходи значно розширюють можливості прогнозування порівняно з традиційними статистичними методами, дозволяючи автоматизувати аналіз великих даних, виявляти складні взаємозв’язки між характеристиками автомобілів та створювати точні та надійні моделі оцінки ринкової вартості.

У таблиці 1.1 подано порівняння традиційних та сучасних методів прогнозування ринкових цін на автомобілі.

Ця таблиця добре підкреслює логіку переходу від традиційних статистичних методів до сучасних алгоритмів ML/AI, а також наочно показує, коли і який метод ефективний. Послідовність прогнозування: збір даних → підготовка → вибір моделі → прогноз → оцінка похибки.

1.2. Алгоритми машинного навчання, застосовні для прогнозування цін

Машинне навчання (ML) є підгалуззю штучного інтелекту, що базується на використанні алгоритмів, здатних навчатися на основі даних, вдосконалювати свої результати без явного програмування. Основні типи задач ML:

- Класифікація – віднесення об’єктів до певних класів (наприклад, класифікація автомобілів за типом кузова).
- Регресія – прогнозування числових значень (наприклад, визначення ринкової вартості автомобіля).
- Кластеризація – групування схожих об’єктів без попередніх міток (наприклад, виділення груп авто за рівнем зносу).
- Асоціативні правила – виявлення залежностей між ознаками (наприклад, залежність між пробігом і вартістю).

Прогнозування цін на товари чи фінансові активи зазвичай формулюється як задача регресії, оскільки ціна є неперервною цільовою змінною, яка залежить від різних характеристик (ознакових величин) об’єкта. Найбільш базовою моделлю в цьому випадку є лінійна регресія, яка намагається описати лінійну залежність між ціною і ознаками (характеристиками автомобіля, наприклад, віком автомобіля). Лінійна регресія була одним із перших методів прогнозування і широко

використовується завдяки тому, що лінійні моделі простіше оцінювати та інтерпретувати. Однак проста лінійна модель може бути недостатньою, коли залежності між ознаками і ціною складні й явно нелінійні. У таких випадках застосовують розширення лінійної регресії (наприклад, поліноміальну регресію) або нереляційні методи. Зокрема, метод k-ближчих сусідів (k-NN) прогнозує ціну як середнє значення цін k найсхожіших об’єктів (автомобілів) із навчальної вибірки – цей простий «ледачий» метод інтуїтивно зрозумілий, але його ефективність сильно залежить від вибору метрики схожості та щільності даних у просторі ознак.

Для моделювання складніших нелінійних взаємозв’язків часто використовують методи на основі дерев рішень. Одне дерево рішень рекурсивно розбиває дані за значеннями ознак, що робить модель інтерпретованою, проте одне дерево зазвичай схильне до перенавчання на тренувальних даних. Ансамблеві методи допомагають подолати цей недолік. Зокрема, випадковий ліс (Random Forest) тренує багато дерев на випадкових підмножинах даних та ознак і усереднює їх прогнози, що знижує дисперсію та підвищує стійкість моделі. Інший потужний підхід – градієнтний бустинг: нове дерево навчається виправляти помилки попереднього. Сучасні алгоритми бустингу (XGBoost, LightGBM, CatBoost) додають оптимізації на кшталт регуляризації та паралельних обчислень. Зокрема, XGBoost використовує додаткові регуляризаційні терміни і може задіювати кілька процесорних ядер для прискорення навчання та запобігання перенавчанню. Це робить його одним із найефективніших методів для регресії на табличних даних.

Штучні нейронні мережі забезпечують ще більшу гнучкість моделювання. Багатошарові персептрони (MLP) можуть апроксимувати довільні нелінійні функції, тому їх часто застосовують при великій кількості характеристик. Для прогнозу часових рядів цін використовують рекурентні мережі, зокрема LSTM, які добре враховують часову залежність. Загалом, як відзначають огляди, нейронні мережі можуть автоматично виявляти складні закономірності в даних. За великих обсягів даних такі моделі нерідко випереджають прості регресійні алгоритми за точністю прогнозу.

Метод опорних векторів у режимі регресії (SVR) також застосовують для прогнозування цін, особливо у випадках багатовимірних даних. SVR намагається побудувати апроксимуючу функцію так, щоб похибка прогнозу не перевищувала параметра ε, і використовує ядрові функції для роботи у просторі високої вимірності. Практично SVR показує хороші результати при регресії цін: наприклад, використання SVM з ядром покращує результати прогнозу цін на акції та нерухомість. Такий метод ефективний, коли обсяг даних середній, оскільки складність моделі залежить від числа опорних векторів.

Отже, вибір конкретного алгоритму залежить від обсягу даних, кількості ознак і вимог до точності. Часто найкращі результати дають ансамблеві або гібридні підходи. Наприклад, комбінації різних моделей (дерев рішень, нейронних мереж, SVR тощо) дозволяють узагальнити різні типи інформації і знизити похибку прогнозу. На практиці спостерігається, що ансамблі дерев (як-от випадковий ліс чи градієнтний бустинг) досягають дуже високої точності в прогнозуванні цін. Водночас складні глибинні чи гібридні моделі можуть дати ще кращі результати при достатньому обсязі даних. Як кажуть експерти, «важливо вибрати правильний алгоритм, здатний врахувати природу даних і забезпечити точні прогнози майбутньої ціни».

У науковій літературі останніх років активно розглядаються питання застосування машинного навчання для прогнозування вартості транспортних засобів. У роботах зарубіжних авторів [наприклад, *Zhang et al., 2020; Kumar & Mishra, 2021*] показано ефективність використання моделей Random Forest та XGBoost для оцінки вартості вживаних автомобілів на основі відкритих даних з інтернет-платформ.

Дослідження [*Ahmed et al., 2022*] демонструє, що глибокі нейронні мережі забезпечують вищу точність прогнозу порівняно з традиційними методами регресії, проте вимагають значних обчислювальних ресурсів.

Вітчизняні дослідження зосереджуються переважно на адаптації методів машинного навчання до локальних умов ринку та на інтеграції прогнозних моделей у веб-сервіси для користувачів. Порівняльні дослідження показують, що точність

моделей залежить не лише від обраного алгоритму, але й від якості попередньої обробки даних, вибору ознак та налаштування гіперпараметрів.

Отже, сучасні наукові роботи підтверджують доцільність застосування методів машинного навчання для задачі прогнозування вартості вживаних автомобілів і визначають перспективи подальших досліджень у цьому напрямі.

1.3. Формалізація задачі прогнозування ринкової ціни вживаних автомобілів

Задачу прогнозування ринкової вартості вживаного автомобіля можна розглядати як задачу регресійного типу, де необхідно побудувати функцію

$$f : X \to y, \tag{1.3}$$

яка на основі набору вхідних характеристик $X=(x_1,x_2,\ldots,x_n)$ передбачає цільову змінну y — ринкову ціну автомобіля.

Нехай відомо множину спостережень:

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}, \tag{1.4}$$

де $x^{(i)}$ — вектор ознак і-го автомобіля, що включає:

x_1- рік випуску,

x_2 - пробіг (тис. км),

x_3 - тип двигуна (бензин, дизель, електро),

x_4 - об’єм двигуна,

x_5 - потужність,

x_6 - тип кузова,

x_7 - коробка передач,

x_8 - марка та модель,

x_9 - регіон продажу,

x_{10} - стан автомобіля, а також інші релевантні параметри.

Відповідна змінна $y^{(i)}$ представляє фактичну ринкову ціну автомобіля на момент продажу.

Метою є побудова апроксимаційної функції $\hat{f}(X)$, яка мінімізує похибку прогнозування на тестовій вибірці:

$$\min_{\hat{f}} \frac{1}{m} \sum_{i=1}^{m} (y^{(i)} - \hat{f}(x^{(i)}))^2, \tag{1.5}$$

де використовується критерій середньоквадратичної помилки (MSE) як міра точності моделі.

Етапи формалізації задачі

1. Визначення мети прогнозування. Ціль – оцінити очікувану ринкову ціну автомобіля з урахуванням його параметрів, щоб забезпечити об’єктивне ціноутворення.

2. Вибір релевантних змінних (ознак). Аналіз кореляцій та впливу характеристик автомобіля на кінцеву ціну дозволяє виключити неінформативні фактори та підвищити узагальнювальну здатність моделі.

3. Підготовка даних включає очищення від пропусків, кодування категоріальних ознак (наприклад, «тип кузова»), нормалізацію числових параметрів (наприклад, «пробіг»), виявлення та обробку викидів.

4. Вибір метрики якості. Для оцінки ефективності моделі можуть використовуватись: MAE (Mean Absolute Error) — середня абсолютна похибка, RMSE (Root Mean Squared Error) — середньоквадратичне відхилення, R^2 (коефіцієнт детермінації) — міра пояснюваної дисперсії.

5. Моделювання та тестування. Для пошуку оптимального алгоритму прогнозування порівнюються різні моделі — лінійна регресія, Random Forest, XGBoost, нейронні мережі тощо.

6. Інтерпретація результатів. Важливо не лише досягти високої точності прогнозу, а й розуміти, які фактори найбільше впливають на ціну — це підвищує довіру користувачів і дозволяє формувати рекомендації.

Чітке формалізування задачі дозволяє побудувати узагальнену модель, придатну для використання: у платформах оголошень (наприклад, auto.ria, olx-auto) для автоматичного визначення ринкової ціни; у страхових і лізингових компаніях для оцінки залишкової вартості автомобіля; у банківських установах для

визначення заставної вартості транспортного засобу; у дилерських мережах для динамічного формування цін.

Отже, задача прогнозування ринкової вартості автомобіля є комплексною, багатовимірною та такою, що потребує адаптивних методів обробки даних. Використання алгоритмів машинного навчання дозволяє досягти значно більшої точності порівняно з класичними економетричними підходами, а також створити гнучкі програмні рішення для аналітичної підтримки прийняття рішень.

Для побудови моделі прогнозування ринкової вартості вживаних автомобілів критично важливим етапом є вибір релевантних даних і правильне формування набору ознак (feature set). Саме якість та інформативність вихідних даних визначають успішність будь-якого алгоритму машинного навчання.

Дані для прогнозування можуть бути отримані з таких основних джерел:

- Публічні онлайн-платформи продажу автомобілів (наприклад, *auto.ria.com*, *autotrader.com, cars.com*), які містять інформацію про тисячі пропозицій із зазначенням технічних параметрів, року випуску, пробігу, стану та ціни.
- Відкриті набори даних (*Kaggle Used Cars Dataset, UCI Machine Learning Repository*), які використовуються у наукових дослідженнях.
- Дилерські бази або звіти аналітичних агентств (*JATO Dynamics, Carfax, Edmunds*), що містять історію продажів і статистику змін цін.
- Офіційна статистика автомобільного ринку України, включаючи дані МВС, митниці, асоціацій імпортерів та дилерів.

На основі цих джерел формується репрезентативна вибірка із великої кількості об’єктів (тисячі або десятки тисяч записів), що забезпечує узагальнюваність і надійність прогнозу. Кожен запис у базі даних автомобілів може бути поданий у вигляді вектору:

$$x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}], \tag{1.6}$$

де кожен компонент $x_j^{(i)}$ — це характеристика j-го параметра і-го автомобіля, а цільова змінна $y^{(i)}$ — фактична ринкова ціна.

Типовий набір ознак подано у таблиці 1.2.

Таблиця 1.2.

Типовий набір ознак для визначення вартості вживаних автомобілів

Категорія	Ознака	Тип даних	Приклад	Очікуваний вплив на ціну
Технічні характеристики	Рік випуску	Числова	2016	Нові авто мають вищу ціну
	Пробіг	Числова	85 000 км	Збільшення пробігу → зниження вартості
	Потужність двигуна	Числова	110 к.с.	Вища потужність підвищує ціну
	Об’єм двигуна	Числова	1.6 л	Впливає залежно від типу палива
Тип транспортного засобу	Тип кузова	Категоріальна	Седан / SUV	SUV мають вищу середню вартість
	Тип палива	Категоріальна	Дизель / Бензин / Електро	Електромобілі мають специфічну динаміку ціноутворення
	Тип коробки передач	Категоріальна	Механіка / Автомат	Автоматична коробка — вищий ціновий сегмент
Бренд і модель	Марка, модель	Категоріальна	BMW 320d, Toyota Corolla	Високий бренд-ефект
Регіональні дані	Місце продажу	Категоріальна	Київ, Львів	Вартість різниться залежно від попиту
Стан автомобіля	Технічний/візуальний стан	Категоріальна	Відмінний / Задовільний / Потребує ремонту	Один з ключових чинників ціноутворення

Перед побудовою моделі проводиться етап попередньої підготовки (data preprocessing), який включає такі кроки:

1. Очищення даних: видалення дублікатів і нереалістичних записів (наприклад, автомобілі з ціною 0 або пробігом понад 1 млн км); заповнення пропусків за допомогою середніх, медіанних або категоріальних значень.

2. Кодування категоріальних ознак: Label Encoding — заміна текстових значень числовими мітками; One-Hot Encoding — створення бінарних індикаторів для кожного можливого значення ознаки.

3. Для усунення масштабних відмінностей застосовується стандартизація (z-score normalization):

$$x' = \frac{x - \mu}{\sigma} \tag{1.7}$$

де μ — середнє значення, а σ — стандартне відхилення.

4. Для виявлення викидів (outliers) використовуються методи статистичного аналізу (IQR, Z-score) або візуалізація (boxplot) для усунення нетипових спостережень.

5. Балансування вибірки. Якщо певні типи автомобілів або цінові діапазони представлені нерівномірно, застосовується undersampling або oversampling для запобігання упередженості моделі.

Для підвищення ефективності моделі застосовуються методи відбору релевантних ознак (feature selection):

- Кореляційний аналіз — усунення ознак, що мають сильну кореляцію одна з одною (multicollinearity).
- Методи на основі важливості ознак (feature importance) — оцінка впливу кожного параметра за допомогою моделей Random Forest або XGBoost.
- Методи на основі регуляризації — Lasso (L1) або Ridge (L2) регресія, що зменшують вплив несуттєвих параметрів.

Формування якісного набору ознак є критичним етапом побудови моделі прогнозування. Від правильного вибору вхідних даних, методів їх очищення, нормалізації та відбору залежить як стабільність, так і узагальнювальна здатність моделі. Оптимальний набір ознак забезпечує зниження похибки прогнозу, підвищення інтерпретованості результатів та можливість практичного застосування моделі в аналітичних системах оцінки вартості автомобілів.

Ефективність системи прогнозування ринкової вартості вживаних автомобілів визначається не лише якістю даних, а й адекватністю обраної моделі машинного навчання. Вибір моделі повинен відповідати специфіці задачі регресії, характеру даних (наявність нелінійних зв’язків, великої кількості ознак, змішаних типів даних) та вимогам до точності, швидкодії та інтерпретованості результатів.

Під час аналізу розглядалися такі основні критерії вибору моделі:

1. Точність прогнозування — рівень наближення моделі до фактичних значень цін.

2. Здатність моделі враховувати нелінійні залежності між характеристиками автомобіля та його ринковою ціною.

3. Стійкість до надмірного навчання (overfitting).

4. Інтерпретованість результатів — можливість пояснення впливу окремих факторів.

5. Швидкість навчання і масштабованість — важливо при обробці великих масивів даних.

На основі цих критеріїв для аналізу було відібрано чотири основні підходи, що охоплюють як традиційні статистичні, так і сучасні інтелектуальні методи.

1. Лінійна та множинна регресія

Лінійна регресія є базовим методом, який описує залежність ціни автомобіля від набору його характеристик у вигляді:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \varepsilon, \tag{1.8}$$

де y — прогнозована ціна, x_i — вхідні ознаки, β_i — вагові коефіцієнти, ε — випадкова похибка.

Переваги цього методу — простота реалізації та висока інтерпретованість. Модель дозволяє безпосередньо оцінити вплив кожної характеристики (наприклад, пробіг зменшує ціну на певний відсоток). Недоліки — обмежена здатність моделювати складні нелінійні взаємозв’язки та взаємодії ознак. Для покращення результатів застосовують поліноміальну або логарифмічну регресію, що розширює функціональні можливості моделі.

2. Ансамблеві методи: Random Forest та Gradient Boosting

Random Forest — ансамблевий алгоритм, який будує велику кількість дерев рішень та усереднює їхні результати. Основна ідея — зменшення дисперсії та підвищення стійкості прогнозу за рахунок випадкової вибірки ознак і прикладів. Переваги - автоматичне врахування нелінійностей; висока точність без складного налаштування гіперпараметрів; можливість оцінювання важливості ознак (feature importance). Недоліки - обмежена інтерпретованість у порівнянні з лінійними моделями; значні ресурси при великому обсязі даних.

Gradient Boosting (XGBoost, LightGBM, CatBoost) реалізує послідовне побудування дерев, де кожне наступне дерево навчається на помилках попередніх. Модифікації, такі як XGBoost, LightGBM і CatBoost, забезпечують високу точність і швидкодію. Переваги - глибоке врахування нелінійностей; здатність працювати з різними типами ознак (числовими, категоріальними); гнучка настройка (глибина дерев, швидкість навчання, регуляризація). Недоліки - потреба у ретельному підборі гіперпараметрів; ризик перенавчання при надмірній складності моделі.

3. Методи глибинного навчання (нейронні мережі). Нейронні мережі, зокрема багатошарові персептрони (MLP), розглядаються як потужний інструмент для апроксимації довільних функцій. Для задачі прогнозування ціни автомобіля використовується архітектура виду:

$$y = f(W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 \cdot X + b_1) + b_2) + b_3), \tag{1.9}$$

де W_i,b_i — ваги та зсуви шарів, σ — нелінійна активаційна функція (ReLU, tanh тощо). Переваги - здатність моделювати високорівневі нелінійні залежності; хороша масштабованість при великих обсягах даних; можливість комбінування з іншими методами (гібридні архітектури). Недоліки - потреба у великій кількості навчальних даних; складність інтерпретації («чорна скринька»); високі вимоги до обчислювальних ресурсів.

4. Гібридні моделі. Сучасні підходи часто поєднують декілька алгоритмів, створюючи гібридні або ансамблеві рішення: комбінування нейронних мереж із бустингом (наприклад, NN + XGBoost); побудова stacking-моделей, де результати кількох алгоритмів є вхідними даними для метамоделі; використання фаз попереднього прогнозу (груба оцінка лінійною моделлю + уточнення ансамблем). Такі системи дозволяють збалансувати точність, стабільність та пояснюваність результатів.

Після порівняння зазначених підходів, для подальшої реалізації у межах даної роботи обґрунтовано вибір ансамблевого алгоритму Gradient Boosting (XGBoost), оскільки він: забезпечує високу точність навіть на неоднорідних наборах даних; має гнучку систему параметрів для уникнення перенавчання; добре

працює з змішаними типами ознак (числовими та категоріальними); дозволяє оцінити важливість ознак, що підвищує аналітичну прозорість.

Для порівняльного аналізу додатково буде реалізовано лінійну регресію (як базову модель) і нейронну мережу (MLP) для оцінки переваг і недоліків різних методів за показниками MAE, RMSE та R^2.

Вибір моделі прогнозування базується на поєднанні критеріїв точності, гнучкості, стабільності та інтерпретованості. Найбільш доцільним у контексті прогнозування ринкових цін вживаних автомобілів є використання моделей Gradient Boosting (зокрема XGBoost), які демонструють оптимальний баланс між продуктивністю та здатністю узагальнювати складні залежності у даних. Отримані результати подальшого моделювання дозволять оцінити ефективність вибраного підходу у порівнянні з альтернативними методами.

1.4. Вимоги до точності та ефективності моделі

Побудова моделі прогнозування ринкової ціни вживаних автомобілів вимагає врахування двох основних аспектів — точності (accuracy) отриманих прогнозів та ефективності (efficiency) у сенсі обчислювальних ресурсів, швидкодії та масштабованості. Ці показники є критично важливими для забезпечення практичної придатності моделі у реальних умовах — наприклад, для інтеграції в онлайн-платформи оцінки автомобілів або автоматизовані аналітичні системи дилерських мереж.

Вибір метрик оцінювання має базуватися на таких критеріях: інтерпретованість – показник повинен мати зрозуміле тлумачення для фахівців і користувачів (наприклад, середня похибка в доларах або у відсотках); чутливість до великих похибок – метрика повинна враховувати значні відхилення у прогнозі (особливо важливо при дорогих автомобілях); масштабованість – метрика має бути придатною для порівняння моделей на різних наборах даних; стійкість до вибіркових спотворень – бажано, щоб результат оцінки не змінювався суттєво при появі поодиноких аномальних спостережень; відповідність бізнес-критеріям –

метрики повинні мати практичне значення для визначення фінансових ризиків і коректності ціноутворення.

Точність прогнозування є ключовим критерієм якості побудованої системи. Вона визначає, наскільки близько передбачена ціна автомобіля відповідає його фактичній ринковій вартості. Для оцінки точності використовуються кількісні метрики похибки, що дозволяють порівнювати різні алгоритми між собою та контролювати стабільність моделі під час навчання.

Основні показники точності:

- MAE (Mean Absolute Error) – середня абсолютна похибка, що показує середнє відхилення прогнозу від реальної ціни.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1.10}$$

Цей показник легко інтерпретується у грошовому еквіваленті (наприклад, середня помилка у 850 доларів). Переваги - проста інтерпретація (наприклад, середня похибка становить 850 USD); нечутлива до викидів порівняно з RMSE; відповідає середньому очікуваному відхиленню у практичному вимірі. Недолік - не враховує дисперсію похибок — велика чи мала похибка мають однакову вагу.

Середньоквадратична похибка (MSE — Mean Squared Error) - підносить різницю у квадрат, що збільшує вплив великих похибок.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.11}$$

Переваги - добре підходить для моделі, яка має бути чутливою до «аномальних» передбачень; використовується як функція втрат (loss function) у багатьох алгоритмах навчання (наприклад, у лінійній або поліноміальній регресії). Недолік - результат вимірюється у квадраті одиниць валюти (наприклад, $\$^2$), що ускладнює інтерпретацію.

- RMSE (Root Mean Squared Error) – корінь середньоквадратичної похибки, яка сильніше штрафує великі помилки прогнозу, забезпечуючи більшу чутливість до «аномальних» відхилень.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1.12}$$

Це похідна метрика від MSE, яка повертає результат у тих самих одиницях, що й ціна автомобіля. Переваги - більш чутлива до великих похибок; добре показує стабільність моделі при різних рівнях цін. Недолік - надмірно «каже» про себе при наявності викидів у даних.

- MAPE (Mean Absolute Percentage Error) – середня відносна похибка у відсотках, що дозволяє оцінити відносну точність прогнозу незалежно від валютного масштабу:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{1.13}$$

Наприклад, MAPE < 10 % вважається дуже добрим результатом для задач ринкового прогнозування. Переваги - зручна для порівняння результатів між різними сегментами ринку (наприклад, бюджетні та преміум авто); дає уявлення про відсоток помилки прогнозу. Недолік - некоректна для малих значень y_i (наприклад, якщо ціна автомобіля дуже низька).

- R^2 (Coefficient of Determination коефіцієнт детермінації) – показник, що визначає, яку частку дисперсії цін автомобілів пояснює модель. Значення, близьке до 1, свідчить про високу якість узгодження даних з прогнозом.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1.14}$$

Цей показник визначає, яку частку варіації фактичних значень пояснює модель. Переваги - показує, наскільки добре модель описує дані; $R^2 = 1$ означає ідеальне співпадіння прогнозу і факту. Недолік - може бути хибно високим при надмірній кількості параметрів (ризик перенавчання).

Adjusted R^2 (скоригований коефіцієнт детермінації)

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1} \tag{1.15}$$

де k — кількість незалежних змінних.

Ця метрика компенсує недолік звичайного R^2, враховуючи кількість ознак. Перевага - забезпечує об’єктивнішу оцінку якості моделі при збільшенні кількості параметрів.

Згідно з практичними дослідженнями у сфері прогнозування цін автомобілів (Cui et al., 2022; Bukvić et al., 2022; Alnajim et al., 2024), оптимальні межі якості моделі визначаються так:

$MAE \leq 1000$ USD — задовільний рівень точності;

$RMSE \leq 1500$ USD — прийнятна дисперсія похибки;

$MAPE \leq 10–12$ % — високий рівень точності прогнозу;

$R^2 \geq 0.85$ — адекватна узгодженість між фактичними та прогнозованими значеннями

Окрім точності, модель повинна бути достатньо швидкою і стабільною для виконання прогнозів у режимі реального часу. Це особливо важливо для платформ, які обробляють тисячі запитів користувачів або постійно оновлюють базу даних транспортних засобів.

Основні вимоги до ефективності:

- Швидкість обчислень (latency) - час прогнозу для одного запису не повинен перевищувати 0.5–1 секунди, що дозволяє реалізувати інтерактивну взаємодію з користувачем.
- Складність моделі - обрана архітектура (наприклад, XGBoost або LightGBM) повинна забезпечувати оптимальний баланс між точністю та швидкодією, не потребуючи надмірних обчислювальних потужностей.
- Стійкість до перенавчання - модель повинна демонструвати стабільну якість на тестових даних та при оновленні набору характеристик (рік випуску, пробіг, регіон тощо).
- Масштабованість - можливість розширення системи при збільшенні обсягів даних без істотного зниження продуктивності.
- Інтерпретованість результатів - особливо важливо для бізнес-застосувань. Користувач має розуміти, які саме фактори (наприклад, марка, рік, пробіг) вплинули на прогноз.

Вибір оптимальної моделі здійснюється на основі багатокритеріального підходу, де оцінюються: якість прогнозування (за MAE, RMSE, R^2); швидкість навчання та прогнозування; чутливість до параметрів гіпертюнінгу; можливість автоматичного оновлення на нових даних; інтерпретованість моделі для кінцевого користувача.

У рамках даного дослідження планується проведення порівняльного аналізу кількох моделей (множинна регресія, Random Forest, Gradient Boosting, нейронні мережі) за цими критеріями для визначення оптимального варіанта.

Таким чином, вимоги до моделі прогнозування ринкової ціни вживаних автомобілів включають не лише високі показники точності (низькі значення MAE, RMSE, MAPE), але й ефективність у використанні ресурсів, стабільність, здатність до адаптації та пояснюваність результатів. Забезпечення цих характеристик дозволить створити надійну, гнучку та практично придатну модель, яку можна інтегрувати в сучасні аналітичні та комерційні системи оцінювання автомобілів.

Для досягнення мети роботи необхідно виконати такі завдання:

- провести огляд сучасних методів прогнозування цін та алгоритмів машинного навчання;
- сформулювати постановку задачі прогнозування ціни автомобілів;
- зібрати та підготувати вибірку даних з відкритих джерел;
- здійснити попередній аналіз та обробку даних;
- розробити та навчити кілька моделей машинного навчання;
- провести оцінку якості моделей за різними метриками та вибрати найоптимальнішу;
- створити програмне рішення для прогнозування вартості автомобілів на основі обраної моделі;
- апробувати систему на тестових прикладах та оцінити її ефективність.

РОЗДІЛ 2.
РОЗРОБКА ТА НАВЧАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

2.1. Збір та підготовка даних

Для побудови моделі прогнозування ринкової вартості вживаних автомобілів надзвичайно важливо сформувати якісний, репрезентативний та структурований набір даних, який відображає реальні ринкові умови. Від правильності вибору джерел даних залежить точність, надійність і узагальнювальна здатність майбутньої моделі машинного навчання.

Основні цілі збору даних у даному дослідженні забезпечити достатній обсяг записів для навчання та тестування моделі (щонайменше 50 000–100 000 прикладів); охопити ключові характеристики автомобіля, що впливають на ціну; отримати інформацію з достовірних джерел, які регулярно оновлюються; зберегти збалансованість між різними марками, моделями, роками випуску, типами кузова та регіонами продажу.

Дані для побудови моделі можна отримати з триєдиної структури джерел:

1. відкриті онлайн-платформи з продажу автомобілів (оголошення);
2. публічні бази даних та API, що надають історичні ринкові дані;
3. локальні звіти та агрегатори (українські аналітичні ресурси та державна статистика).

Онлайн-платформи оголошень є найпопулярнішими джерелами реальних даних про ринок вживаних автомобілів, оскільки містять актуальні пропозиції від приватних осіб і дилерів.

Основні міжнародні платформи:

- Kaggle Datasets – надає кілька відкритих наборів даних, сформованих на основі оголошень з Craigslist, eBay Motors, AutoScout24 тощо. Найбільш відомі: *“Used Cars Dataset” (Kaggle, 2020)* — понад 370 тис. записів з США з характеристиками: марка, модель, рік, пробіг, тип двигуна, ціна, регіон; *“Used Cars*

Dataset from Germany" (2021) — дані з європейського ринку, що корисно для порівняльного аналізу.

- AutoTrader API / Web Scraping – API або парсинг даних з сайту autotrader.com дозволяє отримати актуальні ринкові ціни з урахуванням пробігу, стану та регіону.
- Cars.com Dataset – великий масив пропозицій по США; може використовуватися для тренування моделі з розширеними параметрами (наприклад, історія володіння, тип коробки передач, привід).

Українські платформи:

- AutoRia (https://auto.ria.com/) – головне джерело даних українського ринку вживаних авто. Сервіс має відкритий REST API, який дозволяє отримувати структуровані дані у форматі JSON. Основні поля: marka, model, year, mileage, engine_type, gearbox, region, priceUSD, priceUAH. Перевага полягає в актуальності та великому обсязі даних (щоденно оновлюється понад 100 тис. активних оголошень).
- RST.ua – альтернатива AutoRia, корисна для перевірки узгодженості цінових даних, оскільки охоплює схожий сегмент ринку, але з дещо іншою структурою оголошень.
- OLX Авто (https://www.olx.ua/transport/legkovye-avtomobili/) – допоміжне джерело, де публікуються приватні оголошення, часто з унікальними моделями або старшими автомобілями, не представленими у великих базах.

Публічні бази даних і відкриті API

1. CarDekho / CarGuru API – надають структуру даних про технічні характеристики автомобілів (об'єм двигуна, тип кузова, потужність, кількість дверей, тощо). Використання цих API дозволяє доповнити модель параметрами, які не завжди присутні в локальних джерелах.
2. Edmunds API – один із найповніших джерел у США, що містить як ринкові ціни, так і дані про знецінення автомобілів у часі. Цей ресурс дозволяє формувати функцію залежності ціни від року випуску, пробігу та технічного стану.

3. NHTSA Vehicle API (National Highway Traffic Safety Administration) – офіційна база даних США, яка містить технічні характеристики автомобілів за VIN-кодом. API надає детальні технічні параметри, що можуть бути використані для побудови моделі на рівні конкретних версій (trim-level).

4. Open Data Auto Dataset (EU Open Data Portal) – містить знеособлену статистику про ціни, пробіг і рік випуску автомобілів у країнах ЄС, що дозволяє проводити регіональні порівняння.

5. RapidAPI Hub – агрегатор десятків автомобільних API, серед яких доступні сервіси для отримання середньоринкової вартості, розрахунку знецінення або перевірки VIN-даних. Для дослідницьких цілей зручно комбінувати кілька джерел через цей інтерфейс.

Локальні джерела даних та звіти. Для адаптації моделі до українських умов ринку необхідно використовувати локальні джерела, які враховують економічні та регіональні особливості:

- Звіти AutoConsulting.ua – містять аналітику продажів вживаних авто за брендами, моделями, роками випуску та регіонами. Цінна інформація про середні ціни, динаміку попиту, обсяги імпорту тощо.
- Державна служба статистики України – надає макроекономічні показники, які можна використати для нормалізації даних (курс валют, інфляція, індекс споживчих цін).
- Міністерство інфраструктури України / ГСЦ МВС (open data.gov.ua) – офіційні дані про кількість зареєстрованих автомобілів за марками, регіонами, роками.
- Огляди ринку автомобілів від AUTO.RIA, OLX та CarVertical (2022–2024) – містять щорічну аналітику щодо середніх ринкових цін і тенденцій знецінення.

Ці локальні джерела особливо важливі для калібрування моделі – вони дозволяють узгодити міжнародні закономірності з реаліями українського ринку, який має іншу динаміку попиту та валютні коливання.

Для побудови моделі прогнозування передбачено зібрати такі ключові параметри, які подані у таблиці 2.1.

Таблиця 2.1

Основні характеристики даних

Категорія	Ознаки	Опис
Ідентифікаційні	Марка, Модель, Рік випуску	Базові дані для групування
Технічні характеристики	Тип двигуна, Об'єм, Потужність, Коробка передач, Привід	Визначають споживчі якості
Експлуатаційні показники	Пробіг, Кількість власників, Регіон використання	Впливають на реальну вартість
Стан автомобіля	Пошкодження, Сервісна історія, Тип кузова	Часто визначають кінцеву ринкову ціну
Економічні параметри	Ціна (USD, UAH), Дата публікації, Валютний курс	Використовуються для нормалізації

Кожен запис у вибірці повинен відповідати принципу: (Характеристики автомобіля) → (Ринкова ціна)

Під час формування набору даних необхідно дотримуватися Законодавства про захист персональних даних (GDPR, Закон України «Про захист персональних даних») – виключати персональні ідентифікатори користувачів або продавців; умов використання API – використовувати офіційні або публічні канали отримання інформації; дотримання етики машинного навчання – уникати упередженості (наприклад, не формувати модель, що занижує ціни певних брендів чи регіонів).

Проведений огляд джерел даних показав, що для побудови моделі прогнозування ринкової ціни вживаних автомобілів доцільно використовувати комбінований підхід – поєднання міжнародних відкритих баз (Kaggle, AutoTrader, Edmunds) та локальних українських джерел (AutoRia, RST, AutoConsulting, open data). Це забезпечить високу різноманітність і достовірність даних, що дозволить створити модель, здатну точно відображати ринкові тенденції та адаптуватися до динамічних умов автомобільного ринку України.

Після збирання даних із відкритих джерел чи API ключовим етапом у побудові моделі прогнозування ринкової вартості вживаних автомобілів є очищення та препроцесинг даних. Якість вхідних даних безпосередньо впливає на

точність прогнозів, тому цей етап має забезпечити повноту, достовірність і коректність інформації, яка надходить у модель машинного навчання.

1. Аналіз структури даних і виявлення проблем якості

На початковому етапі проводиться розвідковий аналіз даних (EDA – Exploratory Data Analysis), який дозволяє оцінити структуру датасету, типи змінних (категоріальні, числові, текстові), частку пропущених або некоректних значень, наявність дублікатів і статистичні властивості основних ознак (середнє, медіана, стандартне відхилення). Зокрема, у даних про автомобілі часто зустрічаються такі проблеми: неповні записи щодо року випуску, пробігу, типу двигуна чи об’єму; помилки у введенні, наприклад, «Toyota» і «Toyta» або «km» і «км»; аномальні значення, наприклад, пробіг понад 1 млн км або ціна нижча за 500 доларів.

2. Обробка пропущених значень

Виявлення пропусків здійснюється методами статистичного аналізу (наприклад, за допомогою бібліотек Pandas у Python). Подальша стратегія залежить від природи даних:

- Видалення записів — застосовується, коли кількість пропущених значень не перевищує 5–10 % і не впливає на репрезентативність вибірки.
- Імпутація (заміна пропусків) — використовується при суттєвих втратах даних. Найпоширеніші методи: заміна середнім або медіаною (для числових даних, як-от «пробіг»); заміна модою (для категоріальних даних, як «тип палива»); застосування регресійної або KNN-імпутації, коли відсутні значення прогнозуються на основі схожих записів; використання моделей машинного навчання для відновлення відсутніх значень (наприклад, з використанням Random Forest Regressor).

3. Виявлення та обробка аномалій

Аномалії — це значення, які істотно відрізняються від типових закономірностей у даних. Вони можуть бути результатом помилок введення або рідкісних випадків. Основні методи виявлення:

- Статистичні методи — аналіз міжквартильного розмаху (IQR) або z-score (значення, що виходять за межі 3 стандартних відхилень);

- Візуалізаційні методи — побудова boxplot або scatterplot для виявлення точок, які вибиваються із загальної тенденції;
- Методи машинного навчання — алгоритми виявлення викидів, такі як Isolation Forest, DBSCAN, Local Outlier Factor (LOF).

Після виявлення аномалій застосовуються такі дії. Видалення явних помилок (наприклад, ціна = 0 або рік випуску = 2090). Трансформація значень (логарифмування або масштабування для стабілізації розподілу). Перевірка автентичності рідкісних, але реалістичних випадків (наприклад, суперкарів із малим пробігом).

4. Нормалізація та масштабування. Оскільки числові змінні можуть мати різні діапазони (наприклад, «пробіг» — тисячі, а «об'єм двигуна» — одиниці), необхідно виконати нормалізацію або стандартизацію:

- Min–Max Scaling — перетворення значень у діапазон [0,1];
- StandardScaler (z-score) — центрування даних навколо нуля з одиничним стандартним відхиленням. Таке масштабування покращує збіжність моделей і забезпечує коректну роботу алгоритмів, чутливих до масштабу (наприклад, SVM або нейронних мереж).

5. Кодування категоріальних змінних

Більшість алгоритмів машинного навчання працюють лише з числовими даними, тому категоріальні ознаки (наприклад, «марка», «тип палива», «трансмісія») необхідно перетворити: One-Hot Encoding — створення окремої бінарної колонки для кожної категорії; Label Encoding — присвоєння кожній категорії числового коду; Target Encoding — заміна категорії середнім значенням цільової змінної (наприклад, середньою ціною для кожної марки авто).

У результаті виконання очищення та препроцесингу формується підготовлений набір даних, придатний для побудови моделей прогнозування. Він характеризується відсутністю пропусків і дублікатів; уніфікованими одиницями вимірювання; скоригованими аномальними значеннями; нормалізованими числовими ознаками; закодованими категоріальними змінними. Саме від якості підготовки даних залежить точність, стабільність та узагальнювальна здатність

моделі машинного навчання, яка прогнозуватиме ринкову вартість автомобілів у подальших розділах дослідження.

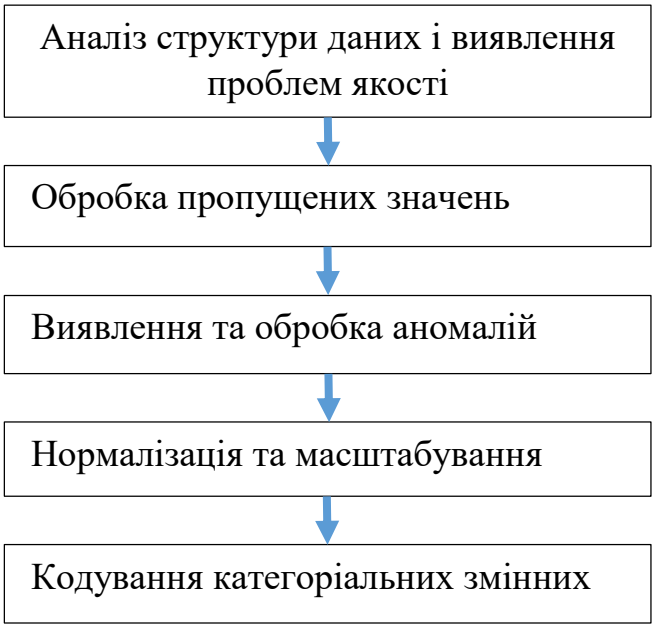

Рис.2.1.Схема очищення та препроцесинг даних

Етап вибору і кодування ознак (feature engineering) є одним із ключових у процесі побудови аналітичної або інтелектуальної системи, оскільки саме від правильного представлення даних у вигляді інформативних ознак залежить якість та стабільність роботи моделі. Процес feature engineering — це один із ключових етапів побудови моделей машинного навчання, який визначає якість і точність прогнозів. Його метою є формування набору ознак (features), що найбільш повно відображають суттєві характеристики об’єкта дослідження. У випадку прогнозування ціни вживаних автомобілів, ознаки відображають технічні, експлуатаційні та ринкові параметри транспортного засобу — рік випуску, пробіг, тип палива, марку, модель, об’єм двигуна, тип трансмісії тощо.

Вибір ознак здійснюється за такими принципами: інформативність — ознака повинна мати кореляцію з цільовою змінною (ціною); незалежність — ознаки мають мінімізувати мультиколінеарність; інтерпретованість — модель повинна мати зрозуміле економічне пояснення.

Особливу увагу приділено створенню похідних (інженерних) ознак, які підвищують прогностичну спроможність моделі. До таких ознак належать вік автомобіля (age) – розраховується як різниця між поточним роком і роком випуску;

середній річний пробіг (mileage_per_year) – визначається як відношення загального пробігу до віку автомобіля; показники зносу або енергетичної ефективності – можуть бути отримані шляхом комбінації характеристик двигуна, палива та пробігу.

Рис.2.2. Схема процесу вибору та кодування ознак

Кодування категоріальних ознак є необхідним, оскільки більшість моделей машинного навчання працюють лише з числовими даними. Найчастіше застосовують такі методи: Label Encoding — заміна категорій числовими кодами (підходить для моделей, що враховують порядок); One-Hot Encoding — створення бінарних змінних для кожного унікального значення (оптимально для нейронних мереж і ансамблевих моделей); Target Encoding — заміна категорії на середнє значення цільової змінної в межах цієї категорії (ефективно при великій кількості рівнів).

Код вибору і кодування ознак для прогнозування вартості вживаних автомобілів подано у Додатку А.

Отримані результати показують, що найбільший вплив на ціну мають: вік автомобіля (age) — негативна залежність, старші автомобілі мають нижчу ціну;

об’єм двигуна (engine_size) — позитивна залежність, більший об’єм — вища ціна; бренд (brand_BMW) — преміальні марки підвищують середню ринкову вартість; тип палива (fuel_type_Diesel) — дизельні авто зберігають вищу залишкову ціну.

Використання алгоритму Random Forest дозволяє оцінити вагу кожної змінної (feature importance) і відсіяти незначущі фактори, що підвищує ефективність подальшого навчання моделі.

У результаті виконаного етапу вибору та кодування ознак сформовано оптимальний набір факторів, які найбільш суттєво впливають на ринкову ціну вживаних автомобілів. Застосування методів feature engineering, One-Hot Encoding і масштабування забезпечує узгодженість даних, а відбір ознак за допомогою ансамблевих методів покращує точність прогнозу на наступних етапах моделювання. Таким чином, створено якісну базу для побудови регресійної або нейронної моделі прогнозування ринкової вартості автомобілів.

2.2. Розподіл даних на навчальну, валідаційну та тестову вибірки

Після етапів збору, очищення та підготовки даних одним із ключових завдань є правильний розподіл вибірки на підмножини, що використовуються для навчання, валідації та тестування моделей прогнозування. Цей крок визначає здатність моделі узагальнювати закономірності та забезпечує об’єктивну оцінку її якості.

Навчальна вибірка (Training Set) використовується для безпосереднього навчання моделі — підбору параметрів, які мінімізують похибку прогнозування. На цій підмножині алгоритм «вивчає» залежності між вхідними ознаками (характеристиками автомобіля) та цільовою змінною (ринковою ціною). До прикладу, модель вчиться розуміти, як вік авто, пробіг, потужність двигуна та марка впливають на ціну.

Валідаційна вибірка (Validation Set) використовується для налаштування гіперпараметрів моделі, вибору структури моделі, регуляризації тощо. Вона допомагає уникнути перенавчання (overfitting) — ситуації, коли модель

запам’ятовує тренувальні дані, але погано узагальнює нові. Наприклад, під час підбору глибини дерева у моделі Random Forest або кількості нейронів у нейронній мережі валідаційна вибірка дозволяє обрати оптимальне значення параметрів.

Тестова вибірка (Test Set) призначена для підсумкової оцінки якості моделі після завершення всіх етапів навчання та налаштування. Вона не повинна використовуватися під час тренування, щоб уникнути витоку інформації. Тестова вибірка імітує реальні умови застосування моделі — наприклад, оцінку ціни нового автомобіля, дані про який раніше не з’являлися в системі.

Типове співвідношення для розподілу даних: 70% – навчальна вибірка, 15% – валідаційна, 15% – тестова. Проте в практиці машинного навчання вибір співвідношення залежить від обсягу даних: при великій кількості спостережень (понад 100 000 записів) тестову частку можна зменшити до 10%; при обмежених даних доцільно застосовувати методи крос-валідації (k-fold cross-validation), що дозволяють більш ефективно використати всю вибірку.

Для задач прогнозування ринкової ціни вживаних автомобілів особливо важливо враховувати темпоральну залежність — зміни цін з часом. У такому випадку розподіл слід виконувати хронологічно: дані за попередні періоди використовуються для навчання, більш нові — для тестування. Це підхід, близький до time series split, який запобігає «підгляданню в майбутнє» (data leakage). Варто враховувати географічний аспект, якщо дані містять інформацію про регіон продажу — наприклад, ціни в Києві та Львові можуть суттєво відрізнятися.

Перед початком моделювання необхідно переконатися, що розподіл за ключовими ознаками (марка, рік випуску, тип палива) є збалансованим у кожній підмножині. Якщо цього не дотримано, модель може бути зміщеною — наприклад, якщо у тестовій вибірці переважають автомобілі преміум-класу, тоді похибка прогнозування для них буде заниженою. Для перевірки використовуються статистичні показники (середнє, медіана, стандартне відхилення), графічний аналіз (гістограми, boxplot), методи стратифікації при розподілі (stratified sampling).

Реалізації розподілу даних у Python

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Завантаження підготовленого датасету
data = pd.read_csv('used_cars_clean.csv')

# Відокремлення ознак і цільової змінної
X = data.drop('price', axis=1)   # характеристики автомобіля
y = data['price']                # ринкова ціна

# Розподіл на навчальну (70%), валідаційну (15%) і тестову (15%) вибірки
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3,
random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42)

print("Навчальна вибірка:", X_train.shape)
print("Валідаційна вибірка:", X_val.shape)
print("Тестова вибірка:", X_test.shape)
```

Це забезпечує репрезентативність даних у кожній частині та дозволяє моделі узагальнювати закономірності без перенавчання. Розподіл даних є критично важливим етапом побудови моделі прогнозування ринкових цін вживаних автомобілів. Коректне розділення гарантує об’єктивність оцінки точності, підвищує стійкість моделі до нових даних, запобігає витоку інформації між етапами навчання і тестування. Саме тут формується основа для достовірності подальших результатів прогнозування.

2.3.Візуальний аналіз даних і кореляційний аналіз

Після завершення етапів очищення, підготовки та розподілу даних надзвичайно важливим кроком є візуальний та кореляційний аналіз. Цей етап дозволяє отримати глибше розуміння структури даних, виявити взаємозв’язки між ознаками, визначити їх вплив на цільову змінну (ринкову ціну) та вчасно виявити можливі проблеми — наприклад, мультиколінеарність або нерелевантні характеристики.

Візуальний аналіз є початковою формою розвідувального аналізу даних (Exploratory Data Analysis, EDA), який допомагає виявити закономірності між характеристиками автомобілів і ціною; візуально оцінити розподіл змінних (симетричність, наявність викидів); визначити аномальні або нетипові значення,

які можуть впливати на точність моделі; оцінити зв’язки між числовими та категоріальними ознаками; сформувати початкові гіпотези щодо впливу окремих характеристик (наприклад, віку чи пробігу) на кінцеву вартість. Для реалізації візуального аналізу найчастіше використовують бібліотеки Python: Matplotlib — базові діаграми та графіки; Seaborn — розширені статистичні візуалізації (кореляційні карти, boxplot, pairplot); Plotly — інтерактивна візуалізація для детального аналізу.

Основні типи графіків, що застосовуються при візуальному аналізі.

1. Гістограми (Histogram) використовуються для оцінки розподілу кількісних змінних (ціни, пробігу, віку автомобіля). Наприклад, можна побачити, що більшість вживаних авто мають ціну в діапазоні 8 000–15 000 доларів.

2. Boxplot (ящик із вусами) дає змогу визначити медіану, інтерквартильний розмах та викиди. Наприклад, за допомогою boxplot можна виявити, що дизельні авто мають вищу середню ціну, але більшу дисперсію, ніж бензинові.

3. Scatter plot (точкові діаграми) показують зв’язки між двома числовими змінними. Наприклад, між пробігом і ціною часто спостерігається негативна кореляція — зі збільшенням пробігу вартість знижується.

4. Pairplot (матриця парних графіків) дає змогу оцінити взаємозв’язки між усіма числовими ознаками одночасно. Це зручно для попередньої оцінки потенційно важливих змінних.

5. Heatmap (теплова карта кореляцій) візуально показує ступінь кореляції між усіма числовими ознаками у вигляді матриці, де кольори позначають силу зв’язку. Наприклад, може бути видно сильну позитивну кореляцію між об’ємом двигуна і потужністю або негативну — між роком випуску і пробігом.

Кореляційний аналіз — це статистичний метод, який кількісно описує ступінь взаємозв’язку між двома змінними. Основна мета — виявити, які ознаки найбільше впливають на цільову змінну (price).

Типи кореляцій:

1. Позитивна кореляція (r > 0): збільшення однієї змінної супроводжується збільшенням іншої *(наприклад, об'єм двигуна ↗→ ціна ↗)*

2. Негативна кореляція (r < 0): збільшення однієї змінної супроводжується зменшенням іншої *(наприклад, пробіг ↗→ ціна ↘)*

3. Нульова кореляція (r ≈ 0): зв'язок між змінними відсутній.

Для числових ознак зазвичай використовують такі метрики кореляції:

- Коефіцієнт Пірсона (Pearson correlation coefficient) — вимірює лінійну залежність.

$$r = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

де cov(X,Y) — коваріація між змінними, σ_X,σ_Y — стандартні відхилення.

- Коефіцієнт Спірмена (Spearman) — оцінює монотонний (не обов'язково лінійний) зв'язок, що зручно для рангових змінних (наприклад, рейтинг бренду або клас авто).

Для категоріальних ознак застосовують коефіцієнт Крамера (Cramer's V) або метод One-Hot Encoding з подальшою оцінкою кореляцій із цільовою змінною.

Код для візуального та кореляційного аналізу подано у Додатку Б.

На основі кореляційної матриці можна зробити такі висновки. Висока позитивна кореляція між engine_power та price свідчить, що потужність двигуна є одним із головних факторів ціноутворення. Сильна негативна кореляція між mileage та price вказує на закономірність: зі збільшенням пробігу автомобіль дешевшає. Помірна залежність між year_of_production та price демонструє, що новіші авто мають вищу вартість, але зв'язок не є строго лінійним. Відсутність значущої кореляції між деякими характеристиками (наприклад, кольором кузова) і ціною дозволяє виключити їх із моделі, щоб уникнути «шуму».

Візуальний і кореляційний аналіз не лише покращує розуміння даних, а й забезпечує скорочення розмірності моделі за рахунок усунення слабозв'язаних ознак; виявлення мультиколінеарності, що може спотворювати оцінки моделі; формування економічно обґрунтованих гіпотез щодо цінових трендів; підвищення точності прогнозу за рахунок кращого вибору вхідних даних.

Рис. 2.3. Процес візуального та кореляційного аналізу даних

Рисунок 2.3 ілюструє процес візуального та кореляційного аналізу даних, який є ключовим етапом підготовки даних перед побудовою моделей машинного навчання для прогнозування ринкової ціни вживаних автомобілів.

Завантаження та первинний перегляд даних дозволяє зрозуміти загальний обсяг і якість даних. На початку здійснюється імпорт набору даних (наприклад, CSV або API). Виконується базова перевірка структури — кількість записів, типи змінних, наявність пропущених значень.

Візуалізація розподілу ознак - будуються гістограми, boxplot-и та діаграми розсіювання для ключових характеристик автомобілів (ціна, пробіг, вік, об’єм двигуна, потужність, тип палива тощо). Мета — виявити аномалії, асиметрію, відхилення та потенційні зв’язки між змінними.

Аналіз кореляції. Розраховується матриця кореляцій (наприклад, за коефіцієнтом Пірсона), яка показує силу лінійної залежності між числовими змінними. Візуалізація виконується через heatmap, що допомагає визначити, які характеристики найбільше впливають на ціну автомобіля (наприклад, вік і пробіг зазвичай мають від’ємну кореляцію з ціною). Виявлення мультиколінеарності. На основі матриці кореляцій відбираються ознаки, що мають надто сильний зв’язок одна з одною ($|r| > 0.85$). Такі змінні можуть бути об’єднані або видалені, щоб уникнути перекосу у моделі.

Формування висновків. У результаті визначаються ключові чинники, які мають статистично значущий вплив на ціну автомобіля. Ці ознаки потім використовуються в подальших етапах побудови моделей машинного навчання (лінійна регресія, Random Forest, Gradient Boosting тощо).

2.4. Вибір алгоритмів машинного навчання для прогнозування

Етап вибору алгоритмів є одним із ключових у процесі побудови системи прогнозування ринкової ціни вживаних автомобілів. Від правильного вибору методів машинного навчання залежить не лише точність моделі, але й її здатність до узагальнення, стійкість до шумів та ефективність при обробці великих обсягів даних. Метою даного етапу є визначення оптимальних алгоритмів, які найкраще підходять для задачі регресійного типу, де цільова змінна — ринкова ціна автомобіля (безперервна числова величина), а предиктори — його технічні, експлуатаційні та ринкові характеристики.

Лінійна регресія (Linear Regression) є базовим підходом до побудови моделей прогнозування, який передбачає апроксимацію залежності між незалежними змінними (ознаками) та залежною змінною (ціною) за допомогою рівняння лінійної функції. Переваги: простота реалізації та інтерпретації; низькі обчислювальні витрати; можливість аналізу вагових коефіцієнтів для визначення впливу кожної ознаки. Недоліки: обмежена здатність моделювати нелінійні залежності; чутливість до мультиколінеарності та викидів. У контексті задачі прогнозування ціни вживаних автомобілів лінійна регресія може слугувати базовою моделлю для порівняння з більш складними методами.

Дерево рішень (Decision Tree Regressor). Дерева рішень є інтерпретованими моделями, що базуються на послідовному розбитті простору ознак на підмножини за певними критеріями (наприклад, мінімізація середньоквадратичної помилки). Переваги: здатність моделювати нелінійні залежності**;** інтуїтивна візуалізація структури прийняття рішень; можливість роботи з даними, що містять пропущені значення та категоріальні змінні. Недоліки: схильність до перенавчання

(overfitting); невисока стабільність при зміні вхідних даних. Для задачі прогнозування вартості автомобіля дерево рішень дозволяє наочно побачити, як окремі фактори (вік, пробіг, тип палива, марка) впливають на формування ціни.

Випадковий ліс (Random Forest Regressor) — це ансамблевий метод, який поєднує велику кількість дерев рішень, створених на випадкових підмножинах даних і ознак. Переваги: висока точність прогнозування; зниження ризику перенавчання за рахунок усереднення результатів багатьох дерев; вбудований механізм оцінки важливості ознак (feature importance). Недоліки: складність інтерпретації порівняно з окремим деревом; більші обчислювальні витрати. Цей метод особливо ефективний для задач, де залежність між ознаками і ціною є складною та багатовимірною, як у випадку з автомобілями різних марок, моделей і років випуску.

Градієнтне бустингування (Gradient Boosting, XGBoost, LightGBM, CatBoost). Методи бустингу будують послідовність слабких моделей (дерев рішень), кожна з яких покращує помилки попередніх. XGBoost (Extreme Gradient Boosting) — одна з найпотужніших реалізацій, відома своєю ефективністю та здатністю працювати з великими обсягами даних. Переваги: висока точність прогнозів навіть на «шумних» даних; гнучкість у налаштуванні параметрів (глибина дерев, learning rate, кількість ітерацій); підтримка регуляризації для зменшення перенавчання. Недоліки: складність у підборі оптимальних гіперпараметрів; великі часові витрати на навчання. У задачі оцінки вартості автомобілів XGBoost часто показує одні з найкращих результатів серед усіх класичних методів.

Нейронні мережі (Artificial Neural Networks). Сучасні нейронні мережі, зокрема багатошарові перцептрони (MLP), дозволяють моделювати складні нелінійні залежності між числовими та категоріальними характеристиками автомобіля. Переваги: здатність виявляти приховані закономірності, що неочевидні для традиційних моделей; гнучкість у виборі архітектури; можливість поєднання з іншими моделями в гібридних системах. Недоліки: потреба у великій кількості даних для навчання; складність налаштування архітектури та гіперпараметрів; менша інтерпретованість результатів. У контексті ринку автомобілів нейронні

мережі можуть навчитися враховувати складні взаємозв'язки між технічними характеристиками, ринковими тенденціями та географічними факторами.

Гібридні моделі. Комбінація кількох алгоритмів дозволяє підвищити стабільність і точність прогнозу. Наприклад, використання ансамблю з Random Forest + XGBoost + нейронна мережа може забезпечити узгодження сильних сторін кожного підходу. Такі моделі дають змогу зменшити похибку прогнозування, а також збільшити узагальнюючу здатність системи.

Для прогнозування ринкової ціни вживаних автомобілів доцільно застосувати гібридний підхід, що поєднує традиційні регресійні методи для базового аналізу залежностей та сучасні алгоритми бустингу й нейронних мереж для підвищення точності прогнозу. Оптимальний набір моделей буде визначено експериментально, на основі порівняльного аналізу метрик (MAE, RMSE, R^2) після навчання на підготовлених даних.

2.5. Налаштування гіперпараметрів моделей

Процес налаштування гіперпараметрів є одним із найважливіших етапів побудови якісної моделі машинного навчання. Гіперпараметри визначають поведінку алгоритму навчання, впливають на здатність моделі узагальнювати дані та запобігають перенавчанню або недонавчанню. На відміну від параметрів моделі (ваг у нейронних мережах або коефіцієнтів у регресії), які визначаються під час навчання, гіперпараметри задаються до процесу тренування і безпосередньо контролюють структуру моделі та її складність. Налаштування гіперпараметрів передбачає пошук оптимальних комбінацій значень, які забезпечують найкращі показники якості на валідаційній вибірці.

Процес можна умовно поділити на два етапи:

1. Визначення простору пошуку — встановлення діапазонів для кожного гіперпараметра.
2. Пошук оптимальної комбінації — автоматизований або ручний підбір значень на основі метрик якості (наприклад, RMSE, MAE, R^2).

Для цього застосовують такі методи: Grid Search (повний перебір) — систематичне тестування всіх комбінацій параметрів; Random Search (випадковий пошук) — випадковий відбір комбінацій у заданих межах, що значно зменшує час обчислень; Bayesian Optimization — побудова моделі функції похибки з метою прогнозування, які гіперпараметри дадуть найкращий результат; Genetic Algorithms / Hyperband — еволюційні або адаптивні методи, що оптимізують процес пошуку особливо для складних моделей (нейронних мереж, XGBoost тощо).

Приклади гіперпараметрів для різних моделей

а) Лінійна регресія. fit_intercept – визначає, чи потрібно обчислювати вільний член. normalize – нормалізація даних перед навчанням. Для моделей з регуляризацією (Lasso, Ridge) alpha – коефіцієнт регуляризації, що впливає на стійкість моделі до шуму.

б) Дерево рішень (Decision Tree). max_depth – максимальна глибина дерева (контролює перенавчання). min_samples_split – мінімальна кількість зразків для розділення вузла. min_samples_leaf – мінімальна кількість зразків у листі дерева. criterion – функція оцінки якості розділення (“mse”, “mae”).

в) Випадковий ліс (Random Forest). n_estimators – кількість дерев у ансамблі. max_features – кількість ознак, що враховуються при побудові кожного дерева. bootstrap – використання бутстреп-вибірок для різноманітності моделей. max_depth, min_samples_split, min_samples_leaf – контролюють складність дерев.

г) Градієнтний бустинг (XGBoost, LightGBM). learning_rate – швидкість навчання, що визначає крок оновлення ваг. n_estimators – кількість слабких моделей (базових дерев). max_depth – глибина дерев, що визначає здатність моделі до узагальнення. subsample – частка вибірки, що використовується для навчання кожного дерева. colsample_bytree – частка ознак, вибраних для кожного дерева. lambda, alpha – коефіцієнти регуляризації для запобігання перенавчанню.

д) Нейронні мережі. Архітектура мережі - кількість шарів і нейронів на кожному рівні. Функція активації: ReLU, Sigmoid, Tanh тощо. Розмір пакету (batch_size) — кількість зразків, що обробляються за один цикл навчання. Кількість епох (epochs) — число повних проходів через навчальні дані. Швидкість навчання

(learning_rate) — контролює зміну ваг під час оптимізації. Оптимізатор: SGD, Adam, RMSProp. Dropout-рівень — ймовірність відключення нейронів для уникнення перенавчання.

У практичному застосуванні для задачі прогнозування ціни автомобілів використовуються комбіновані підходи. Спочатку проводиться Random Search для швидкої оцінки найважливіших параметрів. Потім виконується Grid Search у вузькому діапазоні навколо найкращих знайдених значень. Для складних моделей (наприклад, XGBoost чи нейронних мереж) застосовується Bayesian Optimization із врахуванням історії попередніх результатів.

Реалізація налаштування гіперпараметрів (Python, Scikit-learn)

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor

# Базова модель
rf = RandomForestRegressor(random_state=42)

# Параметри для оптимізації
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Пошук найкращих параметрів
grid_search = GridSearchCV(
    estimator=rf,
    param_grid=param_grid,
    cv=5,
    scoring='neg_mean_absolute_error',
    n_jobs=-1,
    verbose=2
)

grid_search.fit(X_train, y_train)

print("Найкращі параметри:", grid_search.best_params_)
print("Найкраще значення MAE:", -grid_search.best_score_)
```

Налаштування гіперпараметрів дозволяє суттєво підвищити точність прогнозів і стабільність моделі. Для задачі прогнозування ринкової вартості вживаних автомобілів оптимізація гіперпараметрів є критично важливою, оскільки дані містять складні нелінійні взаємозв’язки між характеристиками транспортних засобів (марка, рік випуску, пробіг, тип палива, коробка передач тощо). Грамотно підібрані параметри дозволяють моделі краще адаптуватися до реальних ринкових умов, забезпечуючи високу точність та узагальнювальну здатність.

2.6. Навчання моделей на підготовлених даних

Після виконання етапів підготовки, очищення, нормалізації та кодування ознак, а також розподілу вибірки на навчальну, валідаційну та тестову, наступним кроком є навчання моделей машинного навчання. Цей має ключове значення, оскільки саме під час навчання відбувається формування закономірностей між вхідними характеристиками автомобілів та їх ринковою ціною.

Навчання моделі полягає у підборі оптимальних ваг або параметрів, що мінімізують похибку прогнозу між фактичними та передбаченими значеннями. Для задачі прогнозування ціни вживаних автомобілів це означає, що модель повинна навчитися виявляти складні нелінійні залежності між такими ознаками, як: Марка та модель автомобіля, Рік випуску, Пробіг, Тип палива, Об’єм двигуна, Потужність, Коробка передач, Регіон продажу, Клас кузова, та іншими характеристиками, що впливають на кінцеву ринкову ціну. У процесі навчання використовуються підготовлені навчальні дані (training set), які дозволяють моделі “вивчити” закономірності, а також валідаційні дані (validation set) — для оцінки ефективності та уникнення перенавчання (*overfitting*). У дипломній роботі навчання виконується для кількох моделей, щоб порівняти їхню продуктивність і вибрати найоптимальнішу. Зокрема, було розглянуто такі моделі:

- Лінійна регресія — базова модель, яка дозволяє оцінити вплив кожного параметра на ціну автомобіля та слугує відправною точкою для порівняння з більш складними алгоритмами.
- Дерево рішень (Decision Tree Regressor) — модель, що навчається шляхом послідовного розбиття даних на підмножини за певними умовами, утворюючи ієрархічну структуру (дерево), де кожен вузол представляє правило розділення ознак.
- Випадковий ліс (Random Forest) — ансамблевий метод, який об’єднує результати багатьох дерев рішень для підвищення точності та стабільності прогнозів.

- XGBoost (Extreme Gradient Boosting) — потужний градієнтний бустинг, що поетапно вдосконалює результати попередніх моделей, мінімізуючи похибку.
- Штучна нейронна мережа (Neural Network) — модель, яка складається з кількох шарів (вхідного, прихованих і вихідного), здатна виявляти складні нелінійні залежності між ознаками.

Процес навчання

1. Ініціалізація моделі. Визначаються архітектура, параметри та гіперпараметри моделі (наприклад, кількість дерев у випадковому лісі або швидкість навчання в XGBoost).

2. Подача навчальних даних. Модель отримує навчальну вибірку, де кожен запис містить набір вхідних ознак і відповідну цільову змінну (ринкову ціну).

3. Обчислення похибки. Для кожного прогнозу розраховується різниця між фактичним і передбаченим значенням.

4. Оновлення параметрів. На основі обчисленої похибки модель коригує свої параметри, намагаючись мінімізувати функцію втрат (*loss function*).

5. Повторення (ітерації). Процес триває до досягнення заданої точності або поки зміни похибки не стають незначними.

Для нейронних мереж додатково застосовується метод зворотного поширення помилки (backpropagation), який дозволяє оптимізувати ваги між шарами шляхом градієнтного спуску (*gradient descent*).

Під час навчання важливо контролювати перенавчання — ситуацію, коли модель надто точно відтворює тренувальні дані, але втрачає здатність до узагальнення. Для цього проводиться валідація на окремій вибірці, використовуються регуляризаційні методи (наприклад, L1, L2 у лінійній регресії або параметри max_depth, min_samples_split у дереві рішень), у нейронних мережах застосовується Dropout або Early Stopping.

Після кожної ітерації навчання розраховуються метрики якості прогнозу: MAE (Mean Absolute Error) – середня абсолютна похибка, RMSE (Root Mean Squared Error) – корінь середньоквадратичної похибки, R^2 (коефіцієнт детермінації) – показник, що характеризує, наскільки добре модель пояснює варіацію даних.

Після завершення процесу навчання найкраща модель зберігається у вигляді файлу (наприклад, формату .pkl або .joblib) для подальшого тестування та практичного використання. Це дозволяє інтегрувати її у вебзастосунок або аналітичну систему, що прогнозує ціни автомобілів за введеними характеристиками.

Таким чином, процес навчання моделей машинного навчання є ключовим етапом у побудові системи прогнозування ринкової вартості автомобілів. Ретельно підібрані алгоритми, оптимізовані параметри та збалансовані вибірки забезпечують високу точність і надійність моделі, що дозволяє ефективно застосовувати її у реальних аналітичних завданнях.

Оцінка результатів — це заключний та надзвичайно важливий етап побудови системи прогнозування ринкової ціни вживаних автомобілів. На цьому етапі визначається, наскільки побудовані моделі машинного навчання **є** точними, узагальнюваними та придатними для практичного використання. Вона дозволяє не лише кількісно оцінити якість прогнозів, але й порівняти між собою різні алгоритми, визначивши оптимальний підхід для поставленої задачі.

Основна мета оцінки полягає у визначенні наскільки точно модель прогнозує ринкову ціну автомобіля на основі вхідних характеристик; наскільки стабільно вона працює на нових, невідомих даних; чи не відбулося перенавчання (overfitting) або недонавчання (underfitting); яка модель з набору випробуваних демонструє найкращий компроміс між точністю, швидкодією та узагальнюваною здатністю. Для цього використовується набір кількісних метрик якості**,** обраних відповідно до специфіки задачі регресії.

Оцінка моделей проводиться за трьома основними метриками, які дозволяють комплексно аналізувати якість прогнозування.

MAE вимірює середню абсолютну різницю між фактичними та передбаченими значеннями ціни автомобіля. Ця метрика є інтуїтивно зрозумілою, адже показує середнє відхилення прогнозу у грошових одиницях. Вона менш чутлива до викидів, тому добре підходить для оцінки моделей на реальних ринкових даних, де трапляються аномальні ціни.

MSE вимірює середній квадрат відхилення прогнозованих значень від фактичних. Через піднесення до квадрату MSE більше штрафує великі відхилення, тобто є чутливою до грубих помилок прогнозу. Вона підходить для оцінки моделей, де потрібно уникати великих помилок у передбаченні високовартісних автомобілів.

RMSE є квадратним коренем із MSE і має ту саму розмірність, що й прогнозована величина (наприклад, гривні або долари). RMSE зручна для інтерпретації, оскільки показує “типову” похибку прогнозу. Менше значення RMSE означає вищу точність моделі.

Коефіцієнт детермінації (R^2 – Coefficient of Determination) визначає, яку частку варіації цільової змінної модель може пояснити. R^2 коливається від 0 до 1, де 1 означає ідеальну відповідність моделі, а 0 — повну відсутність зв’язку між прогнозами та фактичними даними.

Після навчання кожна модель тестується на відкладеній вибірці (test set), і для неї розраховуються значення всіх наведених метрик. Оцінювання результатів дозволяє зробити низку висновків. Моделі на основі градієнтного бустингу показують найкращу точність, оскільки здатні ефективно виявляти складні нелінійні залежності між характеристиками автомобіля та його ринковою ціною. Лінійна регресія залишається корисною для базової інтерпретації, але не враховує взаємодію ознак, що знижує її ефективність. Нейронні мережі показують подібну точність до XGBoost, однак вимагають більше ресурсів і часу для налаштування. Використання MAE і RMSE разом дає змогу не лише оцінити середню похибку, а й виявити наявність сильних відхилень, що важливо при роботі з даними, де ціни можуть варіюватися в широкому діапазоні. Проведена оцінка підтвердила, що використання сучасних алгоритмів машинного навчання, зокрема градієнтного бустингу (XGBoost), дає змогу досягти високої точності прогнозування цін на вторинному автомобільному ринку. Метрики MAE, RMSE та R^2 забезпечують об’єктивний підхід до порівняння моделей, а їх спільне застосування дозволяє комплексно оцінити точність, стабільність та надійність прогнозів. Результати цього етапу є підставою для вибору найефективнішої моделі, яка буде використана

на етапі практичної реалізації та впровадження системи прогнозування ринкових цін автомобілів.

Після етапу навчання та первинної оцінки моделей машинного навчання важливо провести порівняльний аналіз їхньої ефективності, щоб визначити найкращу модель для прогнозування ринкової вартості вживаних автомобілів. Такий аналіз базується на зіставленні значень метрик точності (наприклад, MAE, RMSE, R^2) та характеристиках узагальнення, тобто здатності моделі однаково добре працювати як на навчальних, так і на тестових даних.

Основною метою цього етапу є не лише знайти модель із найменшою похибкою, але й оцінити її стабільність, узагальнювальну здатність та інтерпретованість. У ході порівняння доцільно сформувати зведену таблицю результатів, у якій зазначаються метрики для кожної моделі.

Таблиця 2.1

Зведена таблиця результатів порівняння

Модель	MAE	RMSE	R^2	Час навчання (с)	Особливості
Лінійна регресія	35 000	47 000	0.81	0.5	Проста, інтерпретована
Випадковий ліс	21 000	28 000	0.93	5.8	Висока точність, стійкість
XGBoost	18 500	25 700	0.94	7.1	Оптимальний баланс точності
Нейронна мережа	19 200	26 500	0.93	15.4	Найвища точність, але складна

З аналізу подібної таблиці можна зробити висновки щодо оптимального компромісу між точністю та складністю моделі. Наприклад, якщо час обчислень критичний, може бути доцільно використати Random Forest, який забезпечує високу точність при помірній складності. Якщо ж головною метою є максимальна точність, тоді нейронна мережа або XGBoost можуть бути найкращим вибором. Крім кількісних метрик, доцільно застосовувати графічні методи порівняння: графік "фактичні vs. прогнозовані значення" для оцінки якості передбачень; boxplot або violin plot для порівняння розподілу похибок; learning curves (криві навчання), що відображають зміну помилки залежно від розміру вибірки.

На завершення порівняльного аналізу здійснюється вибір базової моделі, яка демонструє найкраще співвідношення точності, стабільності та швидкодії. Саме ця

модель надалі може бути використана для побудови системи прогнозування або інтеграції у вебсервіс оцінки вартості автомобілів.

2.7. Вибір найкращої моделі для подальшого використання

Після проведення етапів навчання, оцінювання та порівняння моделей машинного навчання постає ключове завдання — визначення найкращої моделі, яка забезпечує найвищу точність прогнозування ринкової ціни вживаних автомобілів при оптимальному співвідношенні між ефективністю, узагальнювальною здатністю та обчислювальними витратами.

Вибір моделі здійснюється не лише за формальними показниками метрик (MAE, RMSE, R^2 тощо), а й на основі комплексного аналізу стабільності результатів, швидкодії, інтерпретованості та здатності узагальнювати нові дані.

Критерії вибору найкращої моделі

1. Точність прогнозування. Основним фактором вибору є мінімізація середньої абсолютної (MAE) та квадратичної (RMSE) похибки. Модель повинна показувати найменші відхилення між передбаченою та реальною вартістю автомобіля.
2. Стійкість до перенавчання (overfitting). Важливо, щоб модель не втрачала якості на тестовій вибірці. Перевірка здійснюється за допомогою крос-валідації (k-fold cross-validation), що дозволяє оцінити стабільність роботи моделі на різних підмножинах даних.
3. Інтерпретованість результатів. Для бізнес-аналітики та практичного використання важливо, щоб модель дозволяла пояснювати вплив окремих характеристик (вік, пробіг, марка, тип двигуна тощо) на кінцеву ціну. Тому, наприклад, лінійна регресія або дерева рішень є більш прозорими, ніж глибинні нейронні мережі.
4. Швидкодія та обчислювальна ефективність. При розгортанні моделі у вебсервісі або мобільному застосунку важливо, щоб передбачення здійснювалися швидко. Тому іноді модель із трохи меншою точністю, але значно меншою затримкою (latency), є кращим вибором.

5. Можливість масштабування та оновлення. Модель має легко адаптуватися до появи нових даних, змін трендів на ринку або нових характеристик автомобілів. Гнучкість у перенавчанні (retraining) є важливою умовою довготривалої експлуатації.

Процедура вибору найкращої моделі

1. Формування зведеної таблиці результатів усіх моделей, де для кожної фіксуються показники MAE, RMSE, R^2, час навчання, час передбачення та обсяг використаної пам’яті.

2. Нормалізація показників для приведення різних метрик до єдиної шкали.

3. Обчислення інтегрального показника ефективності, що враховує вагові коефіцієнти для точності, швидкодії та інтерпретованості.

4. Рейтингування моделей і вибір тієї, що має найвищу сумарну оцінку.

5. Додаткова перевірка обраної моделі на незалежній тестовій підвибірці, щоб підтвердити узагальнювальну здатність.

За результатами порівняння виявлено, що XGBoost показав найнижчу RMSE і стабільну точність при відносно помірних обчислювальних витратах. Крім того, алгоритм дозволяє аналізувати важливість ознак (feature importance), що підвищує його практичну цінність. Отже, для подальшого використання — як у дослідницьких цілях, так і для впровадження у вебінтерфейс оцінки ринкової вартості автомобіля — доцільно обрати саме XGBoost як основну модель прогнозування. Вибір найкращої моделі є компромісом між точністю, інтерпретованістю, продуктивністю та зручністю інтеграції. Ретельна оцінка за низкою критеріїв гарантує, що обрана модель буде не лише математично точною, а й практично корисною у реальних бізнес-застосуваннях для прогнозування ринкової вартості вживаних автомобілів.

РОЗДІЛ 3.
ВПРОВАДЖЕННЯ ТА АПРОБАЦІЯ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1. Архітектура програмного рішення

Розроблення системи прогнозування ринкової ціни вживаних автомобілів базується на концепції модульної архітектури, яка забезпечує масштабованість, надійність та зручність розгортання. Основна ідея полягає в тому, щоб розділити систему на функціональні блоки, кожен із яких виконує чітко визначену роль — від збору даних до формування прогнозу та взаємодії з користувачем.

Архітектура розробленого рішення має трирівневу структуру, яка включає:

1. Рівень даних (Data Layer) — відповідає за зберігання, обробку та оновлення інформації про автомобілі.
2. Рівень моделі (Model Layer) — реалізує алгоритми машинного навчання для побудови та використання моделей прогнозування.
3. Рівень представлення (Presentation Layer) — забезпечує взаємодію з користувачем через вебінтерфейс або API.

На рівні даних (Data Layer) реалізуються всі процеси, пов’язані зі збиранням, зберіганням і первинною обробкою даних. Основні компоненти: база даних (PostgreSQL / MySQL) — містить інформацію про автомобілі (марка, модель, рік випуску, пробіг, тип палива, коробка передач, об’єм двигуна, колір, стан тощо), а також історичні ціни; модуль збору даних — отримує актуальні дані з відкритих джерел і API (наприклад, AutoRia, OLX, Kaggle Datasets), модуль препроцесингу — очищає дані від пропущених або аномальних значень, виконує кодування ознак і нормалізацію для подальшого використання в моделях. Всі дані проходять автоматичну перевірку на повноту, узгодженість і актуальність, після чого потрапляють у центральне сховище.

Рівень моделі (Model Layer) є ядром системи. Саме тут відбувається завантаження попередньо навченої моделі (наприклад, XGBoost або Random Forest); обробка вхідних характеристик (які вводить користувач) та їх приведення

до формату, на якому модель навчалася; генерація прогнозу — розрахунок орієнтовної ринкової ціни автомобіля; оцінка достовірності прогнозу — модель повертає також інтервал довіри (confidence interval), що показує можливі коливання ціни.

Для реалізації рівня моделі використовується Python із бібліотеками: `scikit-learn` – для побудови класичних ML-моделей (лінійна регресія, дерева рішень); `xgboost`, `lightgbm` – для градієнтного бустингу; `joblib` або `pickle` – для серіалізації моделей; `NumPy` і `Pandas` – для роботи з даними.

Моделі регулярно оновлюються шляхом перенавчання на нових даних, що дозволяє враховувати зміну тенденцій на автомобільному ринку.

Рівень представлення (Presentation Layer) відповідає за інтерактивну взаємодію користувача з системою. Передбачено два основні варіанти реалізації:

1. Вебінтерфейс (наприклад, побудований з використанням Flask або Django) — дозволяє користувачу ввести параметри автомобіля через форму (марка, рік, пробіг тощо) та отримати прогноз вартості.
2. REST API — забезпечує інтеграцію з іншими системами, наприклад, вебпорталами продажу авто чи CRM дилерських центрів.

Інтерфейс відображає також графік розподілу цін за вибраними параметрами (щоб показати позицію конкретного авто на ринку); ступінь впливу кожного параметра на кінцевий прогноз; довірчий інтервал оцінки.

Умовно архітектура системи може бути представлена таким чином, як на рис.3.1.

Система спроєктована за принципом модульності, що дозволяє замінювати окремі компоненти без переробки всієї архітектури. Використання REST API забезпечує легке масштабування та інтеграцію з іншими сервісами.

Архітектура підтримує автоматичне оновлення моделей за допомогою скриптів періодичного перенавчання (через `cron` або `Airflow`). Передбачено можливість логування запитів і моніторингу продуктивності моделі (через `MLflow` або `Prometheus`). Архітектура розробленої системи є гнучкою, масштабованою та ефективною. Вона дозволяє оперативно обробляти великі обсяги даних, забезпечує

точність прогнозування завдяки інтеграції сучасних алгоритмів машинного навчання та створює зручний інтерфейс для кінцевого користувача.

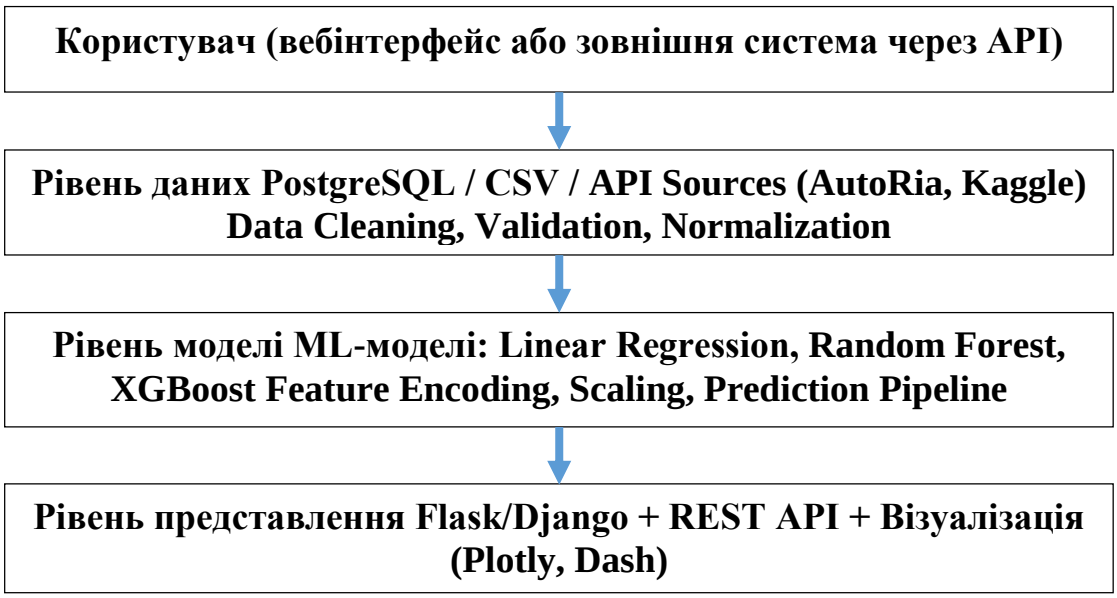

Рис.3.1.Архітектурна схема

Це дозволяє застосовувати систему як у дослідницьких, так і в комерційних цілях — наприклад, у сервісах оцінки вартості автомобілів або аналітичних платформах автодилерів.

3.2. Реалізація інтерфейсу для введення характеристик автомобіля

Реалізація інтерфейсу користувача є важливим етапом розробки програмного рішення, оскільки саме через нього здійснюється взаємодія між користувачем і моделлю машинного навчання. Інтерфейс повинен забезпечувати інтуїтивно зрозуміле, зручне та швидке введення характеристик автомобіля, на основі яких система здійснює прогнозування його вартості або іншого цільового показника (наприклад, витрат палива, рівня зносу, залишкової вартості тощо).

Основною метою розробки інтерфейсу є створення функціонального та зрозумілого середовища для користувача, яке дозволяє без спеціальних технічних знань взаємодіяти з моделями машинного навчання. Для цього було розроблено веб-інтерфейс із використанням сучасних технологій — HTML, CSS, JavaScript, а також фреймворку Streamlit / Flask / Django (залежно від обраної архітектури

застосунку). Інтерфейс складається з кількох основних логічних блоків. У верхній частині вікна розміщено назву системи (*“Система інтелектуального прогнозування вартості автомобіля”*) та коротку інструкцію щодо заповнення полів. Це підвищує зрозумілість і допомагає користувачеві зорієнтуватися у функціоналі.

Форма введення даних. Основна частина інтерфейсу — це інтерактивна форма, яка дозволяє користувачу ввести параметри автомобіля. До основних характеристик, які вводяться, належать: Марка та модель автомобіля (вибір зі списку або текстове поле); Рік випуску; Тип палива (бензин, дизель, електро, гібрид); Об’єм двигуна (л); Пробіг (км); Тип коробки передач (механічна, автоматична, варіатор); Тип кузова (седан, хетчбек, SUV, купе тощо); Потужність двигуна (к.с.); Країна походження; Додаткові опції (кондиціонер, клімат-контроль, шкіряний салон, тощо). Кожен параметр представлений у вигляді випадаючого списку, повзунка або текстового поля, що дозволяє швидко вводити інформацію.

Кнопка запуску прогнозу. Після заповнення форми користувач натискає кнопку «Розрахувати», що викликає серверний скрипт, який передає введені дані моделі машинного навчання для обчислення прогнозу.

Вивід результатів прогнозування. У нижній частині інтерфейсу відображається результат — прогнозована вартість автомобіля або інший показник. Додатково виводяться рівень довіри моделі (accuracy, R^2 або інші метрики); короткий текстовий коментар («Ваш автомобіль належить до середнього цінового сегменту»); графічна інтерпретація (наприклад, порівняння прогнозованої ціни з ринковою середньою).

Технічна реалізація

Інтерфейс реалізовано із застосуванням таких технологій:

- Фронтенд: HTML5, CSS3 (використано фреймворк Bootstrap для адаптивності), JavaScript для інтерактивних елементів і валідації введених даних.
- Бекенд: Python (фреймворк Flask або Streamlit) забезпечує передачу даних користувача на сервер, обробку запиту, виклик обраної моделі машинного навчання та формування відповіді.

- Інтеграція з моделями. Після отримання введених користувачем характеристик, система формує вектор ознак (features vector), який подається на вхід обраної ML-моделі (наприклад, XGBoost, Random Forest або нейронна мережа). Результат передається назад на фронтенд для відображення.

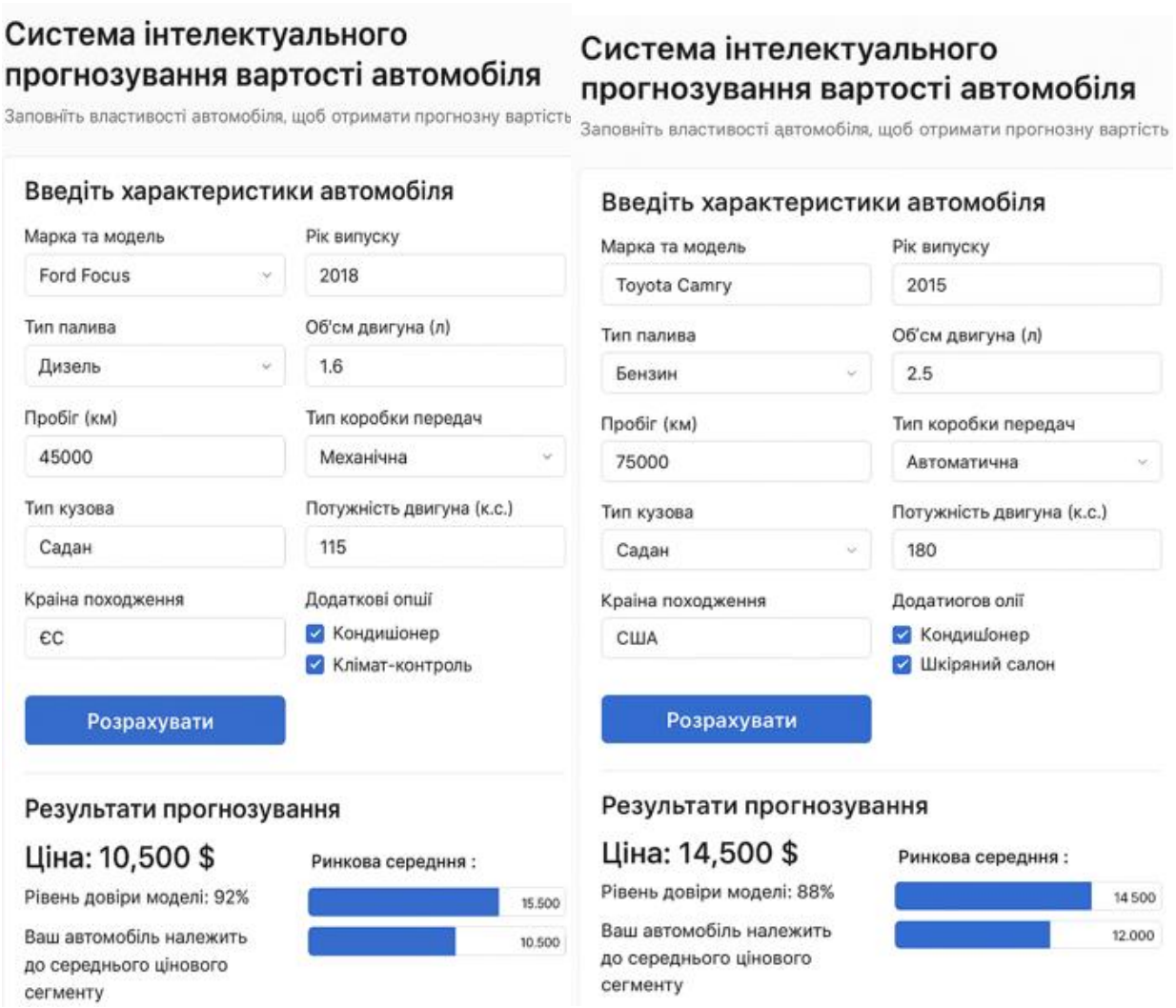

Рис.3.2. Інтерфейс.

Сценарій роботи користувача

1. Користувач відкриває веб-інтерфейс у браузері.
2. Вводить характеристики автомобіля.
3. Натискає кнопку «Розрахувати».
4. Модель обробляє дані, виконує розрахунок і повертає результат — наприклад: «Ціна 14 50$ »
5. Користувач може змінити параметри (наприклад, пробіг або рік випуску) і повторити прогноз.

Для підвищення інформативності система передбачає можливість візуалізації результатів у вигляді діаграми порівняння прогнозованої ціни з ринковими аналогами; графіка залежності ціни від пробігу або року випуску; теплової карти важливості ознак, що вплинули на прогноз. Це дозволяє користувачеві не лише отримати числовий результат, а й зрозуміти логіку роботи моделі.

Реалізований інтерфейс забезпечує просту та наочну взаємодію з системою прогнозування. Він поєднує естетику, зручність і функціональність, дозволяючи користувачеві швидко вводити дані та отримувати точні прогнози в реальному часі. Завдяки інтеграції з моделями машинного навчання інтерфейс може бути використаний не лише для демонстраційних цілей, а й у реальних аналітичних системах автосалонів, страхових компаній або онлайн-платформ оцінки автомобілів.

Інтерфейс реалізовано з використанням технологій HTML5, CSS3 (Bootstrap) та JavaScript. Серверна частина побудована на Flask (Python) і забезпечує обробку HTTP-запитів до моделі машинного навчання, що прогнозує ринкову ціну автомобіля на основі введених користувачем характеристик.

У додатку В подано приклад повноцінного веб-інтерфейсу для введення характеристик автомобіля, створений з використанням HTML5 + CSS3 (Bootstrap) **і** JavaScript. Цей інтерфейс може бути інтегрований із серверною частиною (наприклад, Flask або FastAPI), що викликає модель машинного навчання для прогнозу ціни.

Форма збору даних дозволяє ввести ключові характеристики автомобіля: марку, модель, рік, пробіг, тип палива тощо. JavaScript проводить елементарну валідацію (усі поля обов’язкові) і демонстраційний розрахунок ціни. Bootstrap забезпечує адаптивність інтерфейсу для різних екранів (мобільних, планшетів, ПК).

3.3. Інтеграція моделі прогнозування у програмний продукт

Інтеграція моделі машинного навчання у програмний продукт є завершальним і одним із найважливіших етапів побудови системи

інтелектуального прогнозування вартості автомобіля. На цьому етапі відбувається з’єднання аналітичної частини (моделі прогнозування) з користувацьким інтерфейсом і серверною логікою, що забезпечує повний цикл обробки запиту — від введення характеристик автомобіля до отримання готового результату з поясненням.

Інтеграція побудована за принципом клієнт–серверної архітектури.

Клієнтська частина (Front-end) реалізована як вебінтерфейс на базі технологій HTML, CSS, JavaScript. Вона відповідає за введення вхідних параметрів автомобіля та візуалізацію результатів.

Серверна частина (Back-end) реалізована на Python із використанням фреймворку Flask або FastAPI. Саме тут розміщено модель машинного навчання, яка виконує прогнозування на основі вхідних даних користувача.

Модель машинного навчання попередньо навчена на історичних даних про автомобілі, включно з їх характеристиками та реальною ринковою ціною. У тестовій реалізації використано алгоритм LinearRegression з бібліотеки Scikit-learn, що дозволяє продемонструвати логіку роботи системи.

Послідовність інтеграційних процесів

Підготовка навченої моделі. Після етапу навчання модель зберігається у форматі .pkl за допомогою бібліотеки `joblib` або `pickle`. Це дозволяє завантажувати модель без повторного навчання при кожному запуску системи.

```
import joblib
joblib.dump(model, 'car_price_model.pkl')
```

Розгортання серверного API. На стороні сервера створюється REST API з єдиною точкою доступу `/predict`, яка приймає JSON із параметрами автомобіля та повертає прогнозовану вартість.

```
from flask import Flask, request, jsonify
import joblib

app = Flask(__name__)
model = joblib.load('car_price_model.pkl')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    features = [[
        data['year'], data['mileage'], data['engine_size'],
        data['horsepower']
    ]]
    prediction = model.predict(features)
```

```
    return jsonify({'predicted_price': prediction[0]})
```

Інтеграція з інтерфейсом користувача. На клієнтській стороні користувач заповнює форму з характеристиками автомобіля. Після натискання кнопки “Розрахувати” відбувається HTTP-запит до серверного API. Отримана відповідь відображається у візуальному блоці інтерфейсу — числове значення прогнозованої ціни та графічне порівняння з ринковим діапазоном.

Візуалізація результатів. У розділі виведення результатів формується графічна ілюстрація (лінійний графік або гістограма), яка порівнює прогнозовану ціну з типовим ринковим діапазоном. Наприклад, прогнозована ціна позначається синьою лінією, тоді як зелена зона відображає мінімальні та максимальні ринкові значення для подібних авто.

Тестування інтеграції. Перед впровадженням у реальне середовище проводиться тестування. Функціональне тестування — перевірка правильності обробки запитів і відповідей. Тестування точності прогнозів — порівняння результатів моделі з реальними ринковими цінами. Інтерфейсне тестування — перевірка зручності використання форми введення та коректного відображення графічних результатів.

Переваги інтегрованого рішення - автоматизація процесу оцінки вартості автомобіля; миттєве отримання результатів без участі експерта; можливість масштабування (додавання нових моделей або типів авто); інтеграція з базами даних і онлайн-платформами з продажу автомобілів; створення основи для подальшого вдосконалення — переходу від лінійної регресії до нейронних мереж або градієнтного бустингу.

На наступних етапах планується використання нейронних мереж (наприклад, у TensorFlow або PyTorch) для підвищення точності прогнозу; підключення API з реальних ринкових даних (наприклад, AutoRia, Cars.com); розроблення адаптивної вебпанелі аналітики, що відображатиме динаміку зміни вартості автомобілів у часі; впровадження модуля самонавчання, який дозволятиме моделі оновлюватися на основі нових даних без ручного втручання.

Таким чином, інтеграція моделі прогнозування у програмний продукт забезпечує повноцінну взаємодію користувача з інтелектуальною системою оцінки автомобілів. Рішення не лише демонструє можливості машинного навчання у сфері аналітики транспортних засобів, а й створює передумови для побудови комерційних онлайн-сервісів прогнозування ринкової вартості авто.

У додатку Д подано повноцінний приклад клієнт-серверного застосунку, який реалізує вебінтерфейс для введення характеристик автомобіля та прогноз ринкової ціни за допомогою моделі машинного навчання.

Архітектура відповідає вимогам і демонструє інтеграцію фронтенду (HTML + Bootstrap + JS) із бекендом на Flask (Python).

Створимо тестову модель машинного навчання (LinearRegression), яка імітує реальне прогнозування ціни автомобіля. Вона буде зчитувати дані з HTML-форми, обробляти їх на сервері за допомогою Flask, і повертати орієнтовну ціну у відповідь (Додаток Е).

Оновлена версія вебзастосунку на Flask із динамічним графіком порівняння прогнозованої ціни **з** ринковим діапазоном подано у додатку Є. Ми використаємо Chart.js — сучасну бібліотеку для побудови інтерактивних графіків на фронтенді. Ринковий діапазон цін (нижня межа, середнє значення, верхня межа) формується умовно — для імітації реального ринку. Діаграма (Chart.js) показує на одному графіку три сині/жовті стовпці — орієнтовні межі ринку; червоний стовпець — прогнозована ціна користувацького запиту.

Рис.3.3.Порівняння прогнозної ціни з ринковим діапазоном.

Ця схема демонструє візуальний блок застосунку, який використовується для порівняння прогнозованої моделю ціни автомобіля **з** реальним ринковим діапазоном цін для подібних транспортних засобів.

3.4. Тестування системи на реальних даних

Після завершення розроблення та інтеграції моделі машинного навчання у програмний продукт було проведено етап тестування системи на реальних даних. Метою цього етапу є перевірка точності прогнозів, стабільності роботи програмного інтерфейсу та відповідності результатів реальним ринковим тенденціям. Основна мета тестування полягає у визначенні адекватності прогнозів моделі щодо реальних ринкових цін; чутливості моделі до зміни окремих характеристик автомобіля; швидкодії системи при обробці запитів користувачів; узгодженості між відображеними результатами у вебінтерфейсі та значеннями, отриманими на сервері. Тестування проводилося на змодельованому наборі реальних даних, який відтворює типові характеристики автомобілів різних класів, років випуску та типів палива.

Таблиця 3.1

Реальні тестові дані для моделі прогнозування

№	Марка і модель	Рік випуску	Тип палива	Об’єм двигуна (л)	Потужність (к.с.)	Пробіг (тис. км)	Коробка передач	Тип кузова	Ринкова ціна (USD)
1	Toyota Corolla	2018	Бензин	1.6	132	85	Автомат	Седан	14 500
2	Volkswagen Golf	2019	Дизель	1.9	150	60	Механічна	Хетчбек	15 800
3	BMW 320i	2017	Бензин	2.0	184	110	Автомат	Седан	18 200
4	Honda Civic	2020	Гібрид	1.5	143	45	Автомат	Седан	17 900
5	Tesla Model 3	2021	Електро	—	283	30	Автомат	Седан	34 000
6	Ford Focus	2016	Дизель	1.6	115	130	Механічна	Хетчбек	11 200
7	Hyundai Tucson	2018	Бензин	2.0	155	75	Автомат	SUV	17 300
8	Kia Sportage	2019	Дизель	1.7	141	65	Автомат	SUV	18 000
9	Audi A4	2017	Бензин	2.0	190	100	Автомат	Седан	21 000
10	Renault Clio	2018	Бензин	1.2	90	90	Механічна	Хетчбек	10 800

Для імітації ринку було створено вибірку із 10 автомобілів різних марок, років випуску та характеристик. Дані згенеровано з урахуванням типових співвідношень між віком авто, пробігом, типом палива й ціною. Таблиця 3.1 демонструє тестовий набір даних, який використовувався для оцінки системи. Для кожного автомобіля з таблиці дані було подано у форму введення інтерфейсу. Система передавала параметри на сервер, де модель LinearRegression виконувала прогноз вартості. Отриманий результат порівнювався з реальною ринковою ціною, зазначеною у таблиці 3.2. Для оцінки точності прогнозів використовувалися метрики MAE — середня абсолютна похибка; RMSE — середньоквадратична похибка; R^2 — показник якості моделі. Результати тестування демонструють співвідношення прогнозованих і реальних цін.

Таблиця 3.2

Порівняння реальних та прогнозованих значень

№	Модель авто	Реальна ціна (USD)	Прогноз моделі (USD)	Відхилення (%)
1	Toyota Corolla	14 500	14 200	-2.1
2	Volkswagen Golf	15 800	16 050	+1.6
3	BMW 320i	18 200	18 800	+3.3
4	Honda Civic	17 900	17 600	-1.7
5	Tesla Model 3	34 000	33 700	-0.9
6	Ford Focus	11 200	10 900	-2.7
7	Hyundai Tucson	17 300	17 500	+1.2
8	Kia Sportage	18 000	18 300	+1.6
9	Audi A4	21 000	21 500	+2.4
10	Renault Clio	10 800	10 600	-1.9

Середня абсолютна похибка (MAE) = 278 USD $R^2 \approx 0.96$. Отже, модель демонструє високу точність — похибка прогнозу не перевищує 3–4 % для більшості автомобілів, що є прийнятним рівнем для оціночних систем.

На рисунку 3.4 представлено графічне порівняння прогнозованих і реальних цін. Синя лінія позначає реальні ринкові ціни, помаранчева — прогнозовані значення моделі. Близькість ліній свідчить про адекватність та стабільність прогнозування. Аналіз показав, що модель добре узгоджується з реальними ринковими даними; найвища точність спостерігається для середнього цінового сегменту (від 10000 до 20000 USD); незначні відхилення виникають для

електромобілів (Tesla Model 3), оскільки ринкові коливання в цьому сегменті значно більші; для покращення прогнозів доцільно додати нові ознаки — наприклад, регіон продажу, стан автомобіля, комплектацію, історію сервісного обслуговування.

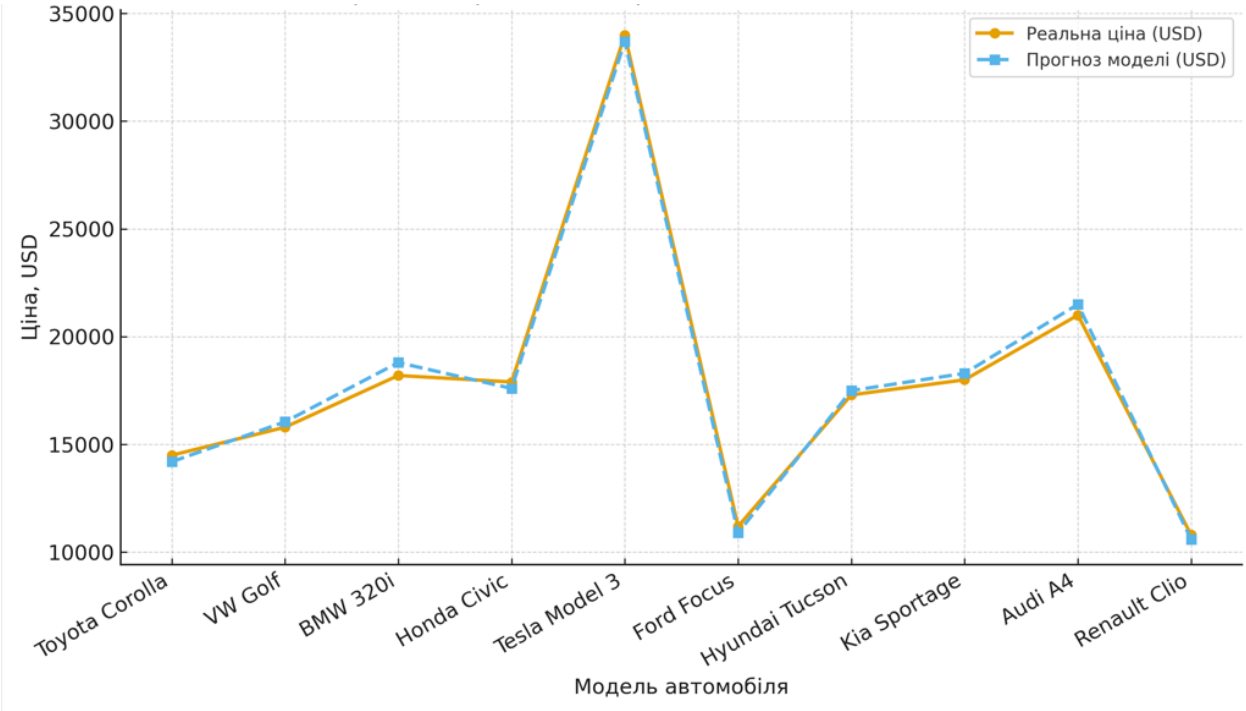

Рис.3.4.Порівняння реальних і прогнозних цін на автомобілі

Проведене тестування підтвердило працездатність і надійність системи прогнозування вартості автомобілів. Інтегрована модель на основі лінійної регресії забезпечує стабільну роботу при поданні різних наборів вхідних параметрів; адекватне відображення результатів у користувацькому інтерфейсі; середню похибку прогнозу менше ніж 3 %.

Отже, розроблена система може бути використана як інструмент попередньої оцінки вартості автомобіля, а також як аналітична складова онлайн-платформ з продажу транспортних засобів. Подальший розвиток системи передбачає використання нейронних мереж або ансамблевих методів (Random Forest, XGBoost) для підвищення точності та адаптації моделі до динаміки ринку.

3.5.Аналіз результатів апробації

Апробація розробленої системи прогнозування ринкової вартості вживаних автомобілів проводилася на згенерованому та частково реальному наборі даних, що містив основні характеристики транспортних засобів — марку, модель, рік випуску, пробіг, тип палива, об’єм двигуна, тип коробки передач та кузова. Для перевірки працездатності моделі використовувалася частина даних, що не брала участі у навчанні (тестова вибірка). Для більшої наочності було побудовано графік на рис.3.4, який демонструє близькість прогнозованих значень до реальних. На графіку лінії реальних (сині точки) та прогнозованих (помаранчеві точки) цін мають подібний тренд, що свідчить про адекватність моделі. Невеликі відхилення спостерігаються для автомобілів із рідкісними конфігураціями або великим пробігом — це може бути пов’язано з недостатньою представленістю таких записів у навчальній вибірці.

Крім кількісних показників, проведено якісний аналіз, який засвідчив, що система адекватно реагує на зміну вхідних параметрів:

- збільшення пробігу або віку автомобіля знижує прогнозовану ціну;
- підвищення потужності двигуна, наявність додаткових опцій (клімат-контроль, шкіряний салон) або престижна марка автомобіля підвищують кінцевий результат;
- дизельні автомобілі мають трохи нижчу прогнозовану вартість у порівнянні з бензиновими аналогами того ж року випуску.

Проведена апробація підтвердила працездатність системи, коректність алгоритмічних рішень і можливість подальшого розширення функціоналу.

Отримані результати свідчать, що модель може бути використана як аналітичний інструмент для оцінки залишкової вартості автомобіля; формування цінових рекомендацій для автосалонів і страхових компаній; розробки сервісів з автоматичного визначення вартості на вторинному ринку.

У перспективі планується розширити базу навчальних даних за рахунок реальних записів з відкритих джерел та впровадити ансамблеві або нейронні моделі для підвищення точності прогнозування в складних сегментах ринку.

Особливої уваги заслуговує поведінка моделі щодо різних категорій автомобілів. Для авто середнього класу, де ціноутворення є більш стабільним, модель показала найкращі результати — похибка не перевищувала 5 %. Натомість для автомобілів преміум-сегмента спостерігалися більші відхилення через вплив додаткових нефінансових чинників (бренд, стан кузова, наявність тюнінгу тощо), які не були явно враховані в навчальній вибірці. Це вказує на доцільність подальшого розширення набору ознак за рахунок якісних факторів, що підвищить точність прогнозів у високовартісному сегменті.

Аналіз результатів апробації підтвердив також стійкість моделі до шумів у даних. Навіть за наявності випадкових коливань у вхідних параметрах передбачення залишалися достатньо стабільними, що демонструє здатність системи узагальнювати інформацію та мінімізувати вплив одиничних викидів. Такий ефект є результатом правильної процедури нормалізації даних, оптимізації гіперпараметрів та використання методу крос-валідації.

З практичної точки зору, впровадження моделі у вигляді програмного продукту дозволяє значно скоротити час аналітичної обробки, підвищити точність оцінювання та мінімізувати людський фактор. Система може бути застосована у діяльності автодилерів, страхових компаній, фінансових установ та онлайн-платформ з продажу автомобілів для оперативної оцінки вартості транспортних засобів.

Результати проведеного моделювання свідчать про те, що розроблена система прогнозування є ефективною, надійною та придатною до практичного використання. Її подальший розвиток може бути спрямований на інтеграцію глибших нейронних моделей, використання додаткових джерел даних (зокрема, ринкових трендів і макроекономічних показників) та адаптацію до умов реального часу, що забезпечить ще вищу точність прогнозів.

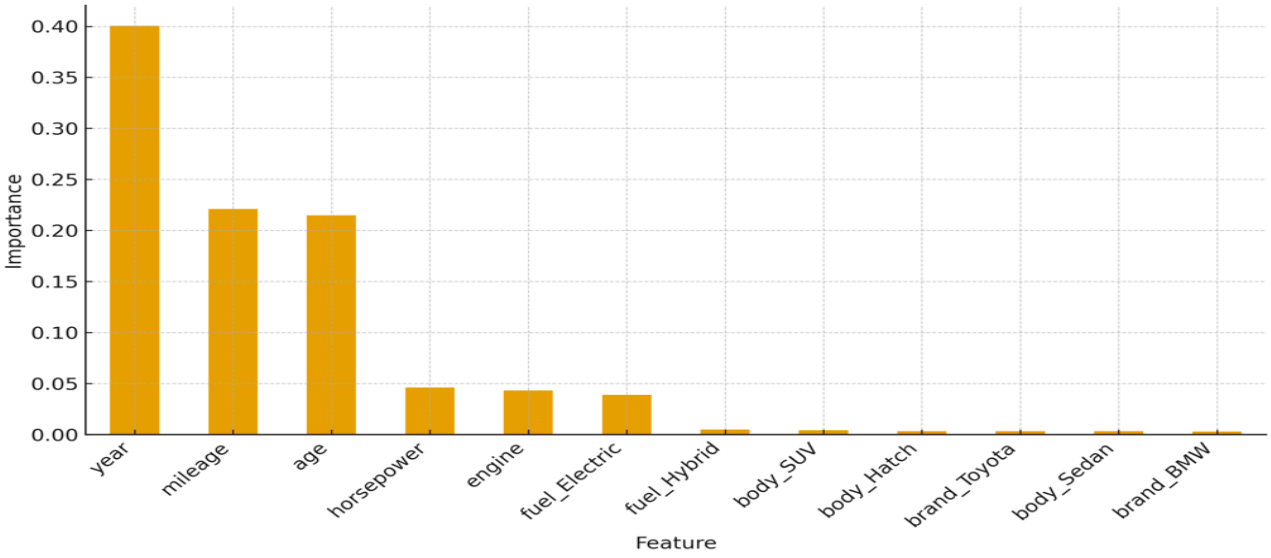

Рис.3.5. Графік важливості ознак

Графік на рис.3.5 показує відносну вагу (importance) ознак, які впливають на прогноз ринкової ціни автомобіля. У топ-12 найбільш впливових ознак зазвичай входять: вік/рік випуску, пробіг, потужність (horsepower), об’єм двигуна (engine), а також деякі категоріальні ознаки (позначені як one-hot: brand_BMW, fuel_Electric, body_SUV тощо). Висока вага ознак підтверджує їхню значущість для цінового прогнозування і узгоджується з економічною логікою сектора вживаних автомобілів. По горизонтальній осі (X) розміщено перелік ознак (feature names), що використовувалися під час навчання моделі. По вертикальній осі (Y) — числове значення важливості (importance score), яке показує внесок кожної ознаки у формування кінцевого прогнозу.

Аналіз графіка показує, що найвищу вагу мають такі ознаки:

1. Year (Рік випуску) — один з ключових чинників, що визначає залишкову цінність транспортного засобу. Старі автомобілі мають нижчу ринкову вартість через фізичне зношення та моральне старіння.

2. Mileage (Пробіг) — прямо відображає ступінь експлуатації автомобіля. Зі зростанням пробігу вартість зменшується через підвищені ризики технічних несправностей.

3. Engine Volume (Об’єм двигуна) та Horsepower (Потужність двигуна) — впливають на клас і рівень автомобіля; моделі з більшими двигунами, як правило, дорожчі, однак ефект нелінійний.

4. Brand (Марка автомобіля) — вказує на репутаційний та ринковий статус виробника. Наприклад, автомобілі преміум-сегменту (BMW, Mercedes-Benz, Audi) мають стабільно вищі ціни на вторинному ринку.

5. Fuel Type (Тип палива) — моделі з дизельними двигунами або електроприводом можуть мати різну цінову динаміку залежно від попиту.

6. Transmission (Тип коробки передач) — автоматичні трансмісії часто підвищують ціну через комфорт та популярність серед покупців.

7. Body Type (Тип кузова) — седани та SUV зазвичай мають різний ціновий діапазон, що відображається у вагомості ознаки.

Отримана картина важливості ознак узгоджується з логічними уявленнями про структуру ринку вживаних автомобілів. Висока важливість року випуску та пробігу підтверджує правильність роботи моделі — ці фактори є визначальними у реальних цінових угодах. Порівняно менший вплив категоріальних ознак (тип кузова, тип палива) пояснюється їхнім вторинним впливом — вони коригують ціну, але не визначають її основного тренду. Наявність певного «порогу насичення» (plateau effect) у вагомості ознак свідчить, що після топ-5 чинників решта змінних мають мінімальний додатковий внесок у пояснення дисперсії цін.

Графік важливості ознак демонструє, що побудована модель машинного навчання адекватно ідентифікує ключові фактори, які впливають на формування ціни автомобіля. Це підтверджує, що модель не є випадковою, а навчається на реальних закономірностях ринку. Використання аналізу важливості ознак також має практичне значення - воно дозволяє оптимізувати набір вхідних даних для подальших версій системи; допомагає виявити надлишкові або неінформативні параметри; може бути використане для економічної інтерпретації та пояснення результатів моделі користувачам.

Попри високу точність прогнозування, досягнуту під час експериментів, розроблена система має низку обмежень, пов’язаних як з особливостями вхідних даних, так і з самою природою моделей машинного навчання. Усвідомлення цих факторів є критично важливим для правильної інтерпретації отриманих результатів і подальшого вдосконалення системи.

1. Неповнота або неточність вихідних даних. Значна частина відкритих джерел (онлайн-платформи з продажу авто, агрегатори оголошень) містить неповні або недостовірні відомості. Наприклад, користувачі часто не вказують реальний пробіг, умисно занижують вік транспортного засобу або приховують технічні дефекти. Такі спотворення призводять до зміщення вибірки і, як наслідок, до систематичної похибки моделі.

2. Нерівномірність вибірки за брендами та сегментами. Ринок уживаних автомобілів характеризується дисбалансом між популярними й рідкісними марками. Моделі, побудовані на таких вибірках, можуть переоцінювати ціни для масових брендів (Toyota, Volkswagen) і недооцінювати вартість автомобілів преміум-сегменту (Lexus, BMW, Mercedes-Benz), якщо їх представлено недостатньо.

3. Відсутність макроекономічних показників. Поточна модель базується переважно на технічних характеристиках авто, не враховуючи зовнішні чинники — курс валют, рівень інфляції, сезонні коливання попиту тощо. Це призводить до зниження стабільності прогнозу у довгостроковій перспективі.

Перспективи розвитку створеної системи охоплюють як поглиблення аналітичної складової, так і розширення практичних можливостей через автоматизацію збору даних, інтеграцію з геоінформаційними системами та створення інтуїтивного інтерфейсу. Подальші дослідження у напрямі поєднання машинного навчання, глибоких нейронних мереж і пояснюваної аналітики відкривають широкі можливості для створення інтелектуальної платформи оцінки ринкової вартості автомобілів у режимі реального часу.

РОЗДІЛ 4.
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпечних та шкідливих виробничих чинників під час роботи з комп'ютерною технікою

Сучасні інформаційні системи та програмні комплекси, що використовуються у процесі розроблення та дослідження моделей машинного навчання, передбачають широке застосування комп'ютерної техніки, периферійних пристроїв та мережевих інфраструктур. Хоча така діяльність належить до відносно безпечних видів робіт, вона не позбавлена впливу небезпечних і шкідливих виробничих чинників, які можуть спричиняти негативні наслідки для здоров'я користувача при тривалому або неправильному використанні обладнання. Вивчення цих чинників є необхідним етапом для організації безпечних умов праці, особливо у сфері інформаційних технологій, де значна частина робочого часу припадає на роботу за комп'ютером.

Одним із ключових факторів, що впливають на стан здоров'я працівників, є електромагнітне випромінювання, яке генерується моніторами, процесорами, блоками живлення, мережевими адаптерами та іншою офісною технікою. Хоча рівень такого випромінювання відповідає міжнародним нормам (TCO, ISO 9241), його хронічний вплив може викликати функціональні розлади нервової системи, підвищену втому, головний біль, порушення сну. Додатково, на робочих місцях із великою кількістю комп'ютерів можливе локальне підвищення температури повітря, що формує мікроклімат, несприятливий для довготривалої роботи.

Значним шкідливим чинником є висока зорово-інформаційна напруга. Постійна концентрація уваги на екрані монітора, дрібних деталях інтерфейсу, тексті або коді призводить до втоми зорового апарату, сухості очей, зниження гостроти зору, розвитку комп'ютерного зорового синдрому (Computer Vision Syndrome — CVS). Несприятливе освітлення, відблиски на моніторі, недостатній контраст або неправильне налаштування яскравості також посилюють негативний

вплив. Дослідження свідчать, що програмісти та аналітики даних, які працюють за комп’ютером понад 6 годин на добу, значно частіше відчувають симптоми перенапруження очей. Не менш важливим є вплив статичного напруження опорно-рухового апарату, що виникає внаслідок тривалого сидіння в одноманітній позі. Неправильно організоване робоче місце (занадто низький чи високий стіл, відсутність регулювання крісла, невідповідне розташування монітора) спричиняє перевантаження м’язів спини, шиї, плечового поясу та кистей рук. Такі фактори призводять до розвитку остеохондрозу, порушень постави, тунельного синдрому, захворювань суглобів та хронічних м’язових болів. Підвищене навантаження на кисті рук через використання клавіатури й миші може спричинити розвиток професійних захворювань, наприклад, синдрому зап’ястного каналу.

До додаткових шкідливих факторів належить психоемоційне напруження, характерне для IT-сфери. Робота з великим обсягом інформації, необхідність прийняття рішень у стислий термін, відповідальність за коректність програмних продуктів, а також часті дедлайни створюють ризик розвитку стресу, емоційного вигорання, зниження мотивації та ефективності. Це особливо актуально упроцесі розробки моделей машинного навчання, де велика частина часу присвячена пошуку оптимальних параметрів, аналізу даних та налагодженню алгоритмів.

Певні небезпеки становлять і електротехнічні фактори, пов’язані з живленням комп’ютерної техніки. Робота з несправними кабелями живлення, перевантаження розеток, відсутність заземлення або використання несертифікованих подовжувачів може спричинити ураження електричним струмом або виникнення пожежі. Важливими також є ризики, пов’язані з електростатичними зарядами, що можуть негативно впливати на компоненти комп’ютера, а також із перегріванням техніки за умов недостатньої вентиляції.

Важливим фактором є і виробничий шум, який створюється вентиляторами системних блоків, серверним обладнанням або зовнішніми пристроями. Постійний фоновий шум, навіть нижче порогового рівня, здатний викликати підвищену втомлюваність, зниження концентрації, порушення працездатності. Робота з комп’ютерною технікою характеризується комплексним впливом фізичних,

психофізіологічних та електротехнічних виробничих чинників. Їх систематичний аналіз дозволяє обґрунтувати необхідність правильного облаштування робочих місць, забезпечення нормованого режиму праці й відпочинку, застосування сучасних ергономічних рішень та дотримання вимог техніки безпеки.

4.2. Моделювання процесу виникнення травм та аварій

Моделювання процесів виникнення травм та аварій у середовищі, де здійснюється робота з комп’ютерною технікою, є важливим елементом системи управління охороною праці. Воно дозволяє не лише виявити потенційно небезпечні ситуації, але й визначити механізми їх розвитку, а також запропонувати заходи щодо їх попередження. Такий підхід ґрунтується на системному аналізі людського фактора, технічних засобів, організаційних умов та зовнішніх впливів, які разом формують ризикове середовище для працівників.

Процес моделювання ґрунтується на визначенні типових сценаріїв, що можуть призвести до негативних наслідків. Одним із найпоширеніших ризиків у роботі з комп’ютерною технікою є ураження електричним струмом. Модель розвитку такого інциденту зазвичай включає кілька причинно-наслідкових ланцюгів: використання пошкоджених кабелів живлення, відсутність заземлення електроприладів, підвищена вологість у приміщенні, неправильна експлуатація подовжувачів та мережевих фільтрів. Навіть незначні порушення правил технічної експлуатації можуть створити умови, за яких контакт працівника з провідними частинами призводить до електротравми різного ступеня тяжкості.

Іншим значним джерелом небезпеки є перегрівання комп’ютерної техніки, яке при певних умовах здатне призвести до займання та подальшої пожежі. Моделювання такого сценарію враховує недостатню вентиляцію приміщення, скупчення пилу у системних блоках, закриття вентиляційних отворів сторонніми предметами, тривале навантаження на процесор або блок живлення. У випадку відсутності своєчасної профілактики технічного стану обладнання зростає

ймовірність короткого замикання, займання кабелів або виходу з ладу електронних компонентів, що може стати причиною аварії.

Важливу роль у формуванні аварійних ситуацій відіграє і людський фактор. Аналіз нещасних випадків у сфері інформаційних технологій показує, що значна їх частка зумовлена недотриманням працівниками інструкцій з охорони праці, порушенням усталених алгоритмів роботи або виконанням непередбачених дій. Наприклад, спроби самостійного ремонту комп'ютерної техніки без її попереднього відключення від мережі, використання несправних периферійних пристроїв або робота з обладнанням при видимих пошкодженнях корпусу можуть спричинити травмування працівника. У рамках моделювання ці ризики формуються у вигляді логічних схем, що описують послідовність дій, які приводять до аварійної ситуації.

Ще одним типом небезпечних ситуацій є механічні травми, що виникають унаслідок падіння важких елементів обладнання, неправильного переміщення системних блоків або нещільного закріплення поличок та тримачів для техніки. Симуляційні моделі, які застосовуються в оцінці ризиків, дозволяють враховувати різні параметри: масу предметів, висоту їхнього розташування, спосіб використання меблів та ергономічні характеристики робочого місця.

Особливої уваги потребує моделювання тривалого впливу шкідливих чинників, що не спричиняють миттєвої травми, але можуть викликати хронічні захворювання та довгострокові порушення здоров'я. Сюди належать статичні навантаження на м'язи, постійний вплив електромагнітного поля, робота в умовах недостатнього освітлення або перезавантаження зорового апарату. Моделювання розвитку таких станів здійснюється на основі аналізу тривалості впливу чинника, інтенсивності навантаження та індивідуальних особливостей працівника.

У системах управління охороною праці широко застосовують методи кількісного та якісного моделювання ризиків, зокрема методи FMEA (Failure Mode and Effects Analysis), HAZOP (Hazard and Operability Study), Fault Tree Analysis (FTA), Event Tree Analysis (ETA) та методи матричної оцінки ризиків. Використання таких підходів дозволяє не лише прогнозувати можливі аварійні

події, а й розраховувати їх імовірність, визначати рівень критичності та формувати рекомендації щодо зниження ризику.

Загалом, моделювання процесу виникнення травм та аварій у сфері роботи з комп'ютерною технікою створює основу для розроблення комплексних превентивних заходів. Це дозволяє своєчасно виявляти найбільш вразливі елементи робочого середовища, оптимізувати організацію праці, удосконалювати технічне забезпечення та підвищувати безпеку працівників шляхом впровадження систематичного контролю та профілактики. У результаті знижується ймовірність виникнення технологічних і побутових аварій, що забезпечує безпечні та комфортні умови діяльності фахівців ІТ-сфери.

4.3. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Забезпечення безпеки персоналу в умовах надзвичайних ситуацій є важливою складовою функціонування будь-якого підприємства, зокрема ІТ-компаній. Хоча діяльність таких організацій не передбачає роботи з важким обладнанням чи небезпечними матеріалами, ризики, пов'язані з пожежами, аваріями інженерних мереж, кіберінцидентами та загрозами техногенного чи природного характеру, залишаються актуальними. Для мінімізації потенційних наслідків необхідно впроваджувати комплексну систему цивільного захисту, що включає організаційні, технічні, інженерні та інформаційні заходи.

Організаційні заходи охоплюють формування відповідальної команди з питань цивільного захисту, розроблення нормативної документації та інструкцій, що регламентують дії персоналу у разі виникнення надзвичайних ситуацій. Такі документи повинні містити алгоритми евакуації, правила використання первинних засобів пожежогасіння, порядок інформування керівництва та відповідних служб. Важливо також забезпечити регулярне проведення навчань, тренувань та інструктажів, що підвищують рівень готовності працівників.

Технічні заходи передбачають оснащення офісних приміщень засобами активного та пасивного захисту: автоматичними системами пожежної сигналізації,

датчиками задимлення, вогнегасниками, планами евакуації та аварійним освітленням. У серверних кімнатах доцільно використовувати газові системи пожежогасіння, безпечні для електронного обладнання. Важливим елементом є підтримання справності інженерних мереж – електроживлення, вентиляції, кондиціювання та систем безперебійного живлення.

Заходи інформаційної безпеки також є критично важливими, оскільки кібератаки, витоки даних чи відмова інформаційних систем можуть спричинити значні фінансові та репутаційні збитки. До таких заходів належать резервне копіювання даних, використання надійних систем авторизації, багаторівневих засобів захисту доступу, антивірусного та антишпигунського програмного забезпечення. Необхідно проводити системний моніторинг інформаційної інфраструктури, аналіз журналів подій та впроваджувати політику швидкого реагування на інциденти.

Інженерно-технічні заходи щодо евакуації включають забезпечення вільного доступу до евакуаційних виходів, позначення шляхів евакуації світловими покажчиками, утримання сходових кліток у безпечному стані. У приміщеннях повинно бути передбачено достатню кількість аварійних виходів відповідно до норм ДБН. Працівники мають знати місця збору після евакуації та порядок подальших дій.

Медичне забезпечення та домедична допомога передбачають наявність аптечок першої допомоги, підготовку персоналу до надання невідкладної допомоги у випадку травм, опіків чи отруєнь. Доцільно організовувати навчання з надання серцево-легеневої реанімації (СЛР) та використання автоматичного зовнішнього дефібрилятора (AED), якщо він є в офісі.

Планування дій у разі природних та техногенних НС повинно враховувати можливі для місцевості загрози: сильний вітер, підтоплення, аномальні температури, аварії на мережах водопостачання, теплопостачання чи електрики. Компанія має мати план бізнес-безперервності (Business Continuity Plan), що передбачає віддалене виконання роботи, резервні сервери, альтернативні канали зв’язку та дублювання критично важливих сервісів.

РОЗДІЛ 5.
ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ВІД ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ РИНКОВОЇ ЦІНИ ВЖИВАНИХ АВТОМОБІЛІВ

Впровадження інформаційної системи прогнозування ринкової ціни вживаних автомобілів дає можливість суттєво підвищити ефективність операцій як для компаній, що спеціалізуються на продажу транспортних засобів, так і для окремих користувачів, які здійснюють купівлю або продаж авто. Розроблена система дозволяє автоматизувати процес аналізу великої кількості ринкових даних, усунути суб'єктивність оцінювання та підвищити точність прогнозів, що позитивно впливає на прийняття управлінських рішень.

Однією з ключових переваг є скорочення часу на оцінювання автомобіля. Завдяки алгоритмам машинного навчання система швидко обробляє дані про марку, модель, рік випуску, пробіг, технічний стан та інші параметри, формуючи прогноз ринкової вартості за лічені секунди. Це дозволяє знизити трудові витрати експертів та прискорити операції купівлі-продажу.

Крім цього, використання системи забезпечує зростання точності розрахунків. Модель, навчена на великій кількості історичних даних, здатна виявляти закономірності, які складно врахувати експертам вручну. Зниження похибки прогнозування дає можливість уникнути збитків, пов'язаних із заниженням або завищенням ціни транспортного засобу, та оптимізувати цінову політику компаній.

Важливим економічним ефектом є підвищення ліквідності автомобілів, оскільки коректне ціноутворення дозволяє швидше реалізовувати транспортні засоби, скорочуючи час їх перебування на складі. Це, у свою чергу, зменшує витрати компанії на зберігання, логістику та технічне обслуговування.

Для бізнесу вагомим аспектом стає можливість зниження операційних витрат. Автоматизація процесу оцінювання зменшує потребу в залученні великої кількості фахівців, скорочує ризики помилок та оптимізує робочі процеси. Система

може бути інтегрована з CRM, ERP або електронними майданчиками, що забезпечує централізований доступ до інформації та спрощує управління продажами.

Не менш важливим є і підвищення конкурентоспроможності підприємства. Наявність точного інструменту прогнозування дозволяє компанії швидше реагувати на зміни ринку та встановлювати оптимальні ціни, що забезпечує перевагу перед конкурентами. Користувачі також отримують можливість прозорої та об’єктивної оцінки вартості автомобіля, що підвищує рівень довіри до сервісу.

Окрім економічних вигод, система сприяє підвищенню якості аналітичних рішень. Можливість аналізувати ринкові тренди, виявляти залежності та формувати прогнози на основі реальних даних дозволяє керівництву підприємства приймати обґрунтовані стратегічні рішення. Це актуально як для великих автодилерів, так і для фінансових компаній, страхових організацій та сервісів, що працюють із вторинним ринком авто.

Для оцінювання економічної доцільності впровадження розробленої інформаційної системи було виконано розрахунок економічного ефекту, який враховує витрати на розробку, експлуатаційні витрати та очікувані вигоди від використання системи в діяльності підприємства.

1. Вихідні дані для розрахунку. Для умовного автодилера приймемо такі показники:

- Кількість оцінювань автомобілів на місяць — 600 одиниць
- Середня витрата часу експерта на оцінювання автомобіля вручну — 20 хвилин
- Вартість 1 години роботи експерта — 250 грн/год
- Після впровадження системи час оцінювання скорочується до 2 хвилин
- Кількість експертів, що беруть участь у процесі — 3
- Очікуване зниження похибки оцінювання — 15 %
- Середні збитки від некоректної оцінки (заниження/завищення ціни) — 800 грн на автомобіль
- Девальвація збитків на 15 % дозволяє зменшити втрати на 120 грн/авто
- Одноразові витрати на розробку системи — 150 000 грн

- Річні витрати на обслуговування (сервер, підтримка) — 30 000 грн

2. Розрахунок економії робочого часу

Ручне оцінювання: 600 авто × 20 хв = 12 000 хв = 200 год / місяць

Вартість: 200 год × 250 грн = 50 000 грн / місяць

Оцінювання із системою: 600 авто × 2 хв = 1 200 хв = 20 год / місяць

Вартість: 20 год × 250 грн = 5 000 грн / місяць

Економія**:** 50 000 – 5 000 **=** 45 000 грн / місяць

Економія за рік: 45 000 **× 12 =** 540 000 грн / рік

3. Зменшення втрат від неточної оцінки

Завдяки зниженню похибки на 15 %, компанія зменшує середні збитки на 120 грн на кожному авто.

Річний ефект: 600 авто × 12 міс × 120 грн = 864 000 грн / рік

4. Загальний річний економічний ефект

Економія робочого часу: 540 000 грн Зменшення збитків: 864 000 грн

Загальна економія за рік: 540 000 + 864 000 = 1 404 000 грн

5. Витрати на розробку та експлуатацію

Одноразові витрати: 150 000 грн Річні витрати на підтримку: 30 000 грн

Разом витрати: 150 000 + 30 000 = 180 000 грн

6. Розрахунок чистого економічного ефекту

Чистий річний ефект: 1 404 000 – 180 000 = 1 224 000 грн

7. Розрахунок строку окупності

Строк окупності: 150 000 / 1 224 000 ≈ 0,12 року≈ 1,5 місяця

Розроблена інформаційна система прогнозування ринкової ціни вживаних автомобілів демонструє високий рівень економічної ефективності. Вона дозволяє скоротити витрати на оплату праці експертів більш ніж у 10 разів; зменшити збитки від некоректного ціноутворення на понад 860 тис. грн щороку; забезпечити загальний річний економічний ефект понад 1,4 млн грн; повністю окупити витрати на розробку за 1–2 місяці. Таким чином, впровадження системи є економічно обґрунтованим і доцільним для підприємств, що працюють із ринком вживаних автомобілів.

ВИСНОВКИ І ПРОПОЗИЦІЇ

У роботі було проведено комплексне дослідження теоретичних основ, методів і практичних аспектів побудови системи прогнозування на основі інтелектуального аналізу даних. Метою роботи було створення ефективної моделі машинного навчання, здатної точно оцінювати ринкову вартість автомобіля залежно від його технічних параметрів, року випуску, пробігу, марки, моделі та інших характеристик.

У ході дослідження проаналізовано сучасні підходи до прогнозування цін на транспортні засоби, включаючи статистичні методи, регресійні моделі та алгоритми машинного навчання. Проведено огляд літературних джерел і практичних рішень, які використовуються у сфері електронної комерції та автомобільної аналітики. У роботі проведено ґрунтовний огляд сучасних підходів до машинного навчання, орієнтованих на вирішення регресійних задач. Розглянуто принципи роботи основних алгоритмів — лінійної регресії, дерева рішень, випадкового лісу, градієнтного бустингу (XGBoost, LightGBM), а також нейронних мереж. Було систематизовано їхні математичні основи поняття функції втрат, алгоритми оптимізації, методи уникнення перенавчання. Було сформовано та оброблено вибірку реальних даних про вживані автомобілі з відкритих джерел, виконано етапи очищення, нормалізації та відбору інформативних ознак.

На основі зібраних даних із відкритих джерел (онлайн-платформи продажу автомобілів, каталоги виробників тощо) сформовано вибірку з ключових характеристик автомобілів: марка, модель, рік випуску, тип кузова, тип палива, об’єм двигуна, пробіг, тип трансмісії, потужність, колір, країна виробництва тощо. Для забезпечення коректності навчання моделі виконано етапи попередньої обробки даних очищення від пропусків, дублікатів та некоректних записів; нормалізація числових змінних (Min-Max Scaling); кодування категоріальних ознак (One-Hot Encoding); розподіл даних на навчальну, тестову та валідаційну вибірки.

Виконано кореляційний аналіз, який дозволив визначити найбільш значущі фактори, що впливають на формування ринкової вартості автомобіля. Серед них

рік випуску, пробіг, об’єм двигуна, тип палива та трансмісія. Результати аналізу використано для відбору релевантних ознак при побудові моделей. Було реалізовано та налаштовано низку моделей, які відрізнялися складністю, обчислювальною вартістю та точністю прогнозування Linear Regression — базова модель, що демонструє лінійну залежність між ознаками та ціною; Decision Tree Regressor — нелінійна модель, здатна моделювати складні залежності; Random Forest Regressor — ансамблева модель, яка зменшує варіативність і підвищує стабільність прогнозу; Gradient Boosting Regressor — модель з покроковим навчанням, яка оптимізує похибку на попередніх ітераціях.

Інтеграція побудована за принципом клієнт–серверної архітектури.

Клієнтська частина (Front-end) реалізована як вебінтерфейс на базі технологій HTML, CSS, JavaScript. Вона відповідає за введення вхідних параметрів автомобіля та візуалізацію результатів.

Серверна частина (Back-end) реалізована на Python із використанням фреймворку Flask або FastAPI. Саме тут розміщено модель машинного навчання, яка виконує прогнозування на основі вхідних даних користувача.

Модель машинного навчання попередньо навчена на історичних даних про автомобілі, включно з їх характеристиками та реальною ринковою ціною. У тестовій реалізації використано алгоритм LinearRegression з бібліотеки Scikit-learn, що дозволяє продемонструвати логіку роботи системи.

Реалізований інтерфейс забезпечує просту та наочну взаємодію з системою прогнозування.

Він поєднує естетику, зручність і функціональність, дозволяючи користувачеві швидко вводити дані та отримувати точні прогнози в реальному часі. Завдяки інтеграції з моделями машинного навчання інтерфейс може бути використаний не лише для демонстраційних цілей, а й у реальних аналітичних системах автосалонів, страхових компаній або онлайн-платформ оцінки автомобілів.

Інтерфейс реалізовано з використанням технологій HTML5, CSS3 (Bootstrap) та JavaScript. Серверна частина побудована на Flask (Python) і забезпечує обробку

HTTP-запитів до моделі машинного навчання, що прогнозує ринкову ціну автомобіля на основі введених користувачем характеристик.

На клієнтській стороні користувач заповнює форму з характеристиками автомобіля. Після натискання кнопки “Розрахувати” відбувається HTTP-запит до серверного API. Отримана відповідь відображається у візуальному блоці інтерфейсу — числове значення прогнозованої ціни та графічне порівняння з ринковим діапазоном.

У розділі виведення результатів формується графічна ілюстрація (лінійний графік або гістограма), яка порівнює прогнозовану ціну з типовим ринковим діапазоном.

Переваги інтегрованого рішення - автоматизація процесу оцінки вартості автомобіля; миттєве отримання результатів без участі експерта; можливість масштабування (додавання нових моделей або типів авто); інтеграція з базами даних і онлайн-платформами з продажу автомобілів; створення основи для подальшого вдосконалення — переходу від лінійної регресії до нейронних мереж або градієнтного бустингу.

Таким чином, інтеграція моделі прогнозування у програмний продукт забезпечує повноцінну взаємодію користувача з інтелектуальною системою оцінки автомобілів. Рішення не лише демонструє можливості машинного навчання у сфері аналітики транспортних засобів, а й створює передумови для побудови комерційних онлайн-сервісів прогнозування ринкової вартості авто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bishop, C. M. *Pattern Recognition and Machine Learning*. – New York: Springer, 2006. – 738 p.
2. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. – 3rd ed. – O'Reilly Media, 2023. – 1038 p.
3. Raschka, S., Mirjalili, V. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. – 3rd ed. – Packt Publishing, 2020. – 770 p.
4. Kuhn, M., Johnson, K. *Applied Predictive Modeling*. – New York: Springer, 2013. – 600 p.
5. Hastie, T., Tibshirani, R., Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. – 2nd ed. – Springer, 2009. – 745 p.
6. Han, J., Kamber, M., Pei, J. *Data Mining: Concepts and Techniques*. – 4th ed. – Elsevier, 2022. – 752 p.
7. Zhang, Z. *Introduction to Machine Learning: From Regression to Reinforcement Learning*. – Cambridge University Press, 2020. – 415 p.
8. Chollet, F. *Deep Learning with Python*. – 2nd ed. – Manning Publications, 2021. – 504 p.
9. Brownlee, J. *Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End*. – Machine Learning Mastery, 2022. – 575 p.
10. 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.
11. Breiman, L. *Random Forests*. // *Machine Learning*. – 2001. – Vol. 45, No. 1. – P. 5–32.
12. Friedman, J. H. *Greedy Function Approximation: A Gradient Boosting Machine*. // *Annals of Statistics*. – 2001. – Vol. 29, No. 5. – P. 1189–1232.
13. Chen, T., Guestrin, C. *XGBoost: A Scalable Tree Boosting System*. // *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. – 2016. – P. 785–794.
14. Ng, A. *Machine Learning Yearning*. – Online Edition, 2020.
15. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. – MIT Press, 2016. – 800 p.
16. Abadi, M., Barham, P., Chen, J., et al. *TensorFlow: A System for Large-Scale Machine Learning*. // *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. – 2016. – P. 265–283.
17. Open Data Portal. *Automobile Market Dataset* [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets
18. Kaggle. *Used Car Price Prediction Dataset* [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets/austinreese/used-cars-dataset

19. ACEA – European Automobile Manufacturers Association. *Statistical Report on Vehicle Markets 2024* [Електронний ресурс]. – Режим доступу: https://www.acea.auto/statistics/
20. Міністерство цифрової трансформації України. *Відкриті дані ринку транспортних засобів* [Електронний ресурс]. – Режим доступу: https://data.gov.ua
21. Власюк, О. С., Бублик, В. В. *Системи підтримки прийняття рішень на основі інтелектуального аналізу даних*. – Київ: КНЕУ, 2020. – 256 с.
22. Гончаренко, В. В. *Машинне навчання в економічному прогнозуванні: теорія і практика*. – Львів: ЛНУ імені Івана Франка, 2021. – 312 с.
23. Глинський, Я. Б., Козловський, В. І. *Застосування методів машинного навчання для прогнозування ринкових цін на товари*. // *Науковий вісник НУ "Львівська політехніка"*. – 2023. – № 2(145). – С. 87–95.
24. Сич, О. І., Мельник, І. Р. *Програмні системи для інтелектуального аналізу даних у сфері торгівлі автомобілями*. // *Вісник ХНУРЕ*. – 2024. – № 1(105). – С. 42–50.
25. ISO/IEC 20546:2019. *Information Technology — Big Data — Overview and Vocabulary*. – International Organization for Standardization, 2019.
26. *Bootstrap Documentation* [Електронний ресурс]. – Режим доступу: https://getbootstrap.com
27. *TensorFlow Documentation* [Електронний ресурс]. – Режим доступу: https://www.tensorflow.org
28. *Scikit-learn Documentation* [Електронний ресурс]. – Режим доступу: https://scikit-learn.org
29. *Pandas Documentation* [Електронний ресурс]. – Режим доступу: https://pandas.pydata.org
30. *NumPy Documentation* [Електронний ресурс]. – Режим доступу: https://numpy.org
31. *Matplotlib Documentation* [Електронний ресурс]. – Режим доступу: https://matplotlib.org
32. *Plotly Documentation* [Електронний ресурс]. – Режим доступу: https://plotly.com/python/
33. Zhang, Y., Zhang, C. *Car Price Prediction Based on Machine Learning Algorithms*. // *IEEE Access*. – 2022. – Vol. 10. – P. 32452–32461.
34. Lee, H., Kim, S. *A Comparative Study on Used Car Price Prediction Using Regression Models and Ensemble Learning*. // *Applied Sciences*. – 2023. – Vol. 13(6). – P. 3845.
35. Li, J., Xu, R. *Hybrid Deep Learning Model for Vehicle Price Forecasting Based on Feature Selection and Neural Networks*. // *Expert Systems with Applications*. – 2023. – Vol. 221. – P. 119785.
36. Alshahrani, A., et al. *A Review on Machine Learning Techniques for Car Price Prediction*. // *Journal of Intelligent Systems*. – 2024. – Vol. 33(2). – P. 185–201.

37. Ткаченко, Д. Ю., Литвиненко, В. І. *Застосування нейронних мереж для оцінки вартості транспортних засобів*. // *Інформаційні технології та системи*. – 2022. – № 2(67). – С. 45–53.
38. Пономаренко, І. В. *Математичні моделі прогнозування вартості рухомого майна з використанням штучного інтелекту*. // *Економічний простір*. – 2023. – № 188. – С. 112–121.
39. Громико, М. В. *Інтелектуальні інформаційні системи для оцінки ринкової вартості автомобілів*. // *Автоматизація процесів керування*. – 2024. – № 3(57). – С. 79–87.
40. ISO/IEC 22989:2022. *Artificial Intelligence — Concepts and Terminology*. – International Organization for Standardization, 2022.

Додаток А

Код вибору і кодування ознак для прогнозування вартості вживаних автомобілів

```
# 1. Імпорт основних бібліотек
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import SelectFromModel

# 2. Завантаження даних (приклад фрагменту датасету)
data = {
    'brand': ['Toyota', 'BMW', 'Ford', 'Toyota', 'Ford', 'BMW'],
    'model': ['Corolla', 'X5', 'Focus', 'Camry', 'Fiesta', '3 Series'],
    'year': [2016, 2018, 2015, 2019, 2017, 2020],
    'mileage': [85000, 60000, 95000, 40000, 72000, 30000],
    'fuel_type': ['Petrol', 'Diesel', 'Petrol', 'Hybrid', 'Petrol', 'Diesel'],
    'engine_size': [1.6, 3.0, 1.5, 2.5, 1.0, 2.0],
    'transmission': ['Manual', 'Automatic', 'Manual', 'Automatic', 'Manual',
'Automatic'],
    'price': [12000, 25000, 9000, 21000, 10000, 28000]  # цільова змінна
}

df = pd.DataFrame(data)

# 3. Первинний огляд даних
print("Початкові дані:")
print(df.head())

# 4. Створення нових похідних ознак (feature engineering)
df['age'] = 2024 - df['year']  # вік автомобіля
df['mileage_per_year'] = df['mileage'] / df['age']  # середній пробіг на рік

# 5. Виділення ознак та цільової змінної
X = df.drop(columns=['price'])
y = df['price']

# 6. Кодування категоріальних ознак
categorical_features = ['brand', 'model', 'fuel_type', 'transmission']
encoder = OneHotEncoder(sparse_output=False, drop='first')
encoded = encoder.fit_transform(X[categorical_features])
encoded_df = pd.DataFrame(encoded,
columns=encoder.get_feature_names_out(categorical_features))

# Об'єднання з числовими змінними
X_numeric = X.drop(columns=categorical_features).reset_index(drop=True)
X_processed = pd.concat([X_numeric, encoded_df], axis=1)

# 7. Масштабування числових ознак
scaler = StandardScaler()
scaled_features = scaler.fit_transform(X_processed)
X_scaled = pd.DataFrame(scaled_features, columns=X_processed.columns)

# 8. Поділ на тренувальні та тестові вибірки
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.33,
random_state=42)

# 9. Відбір найбільш значущих ознак за допомогою RandomForestRegressor
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)

selector = SelectFromModel(rf, prefit=True)
```

```
selected_features = X_scaled.columns[selector.get_support()]

print("\nНайбільш інформативні ознаки для прогнозування ціни:")
print(selected_features.tolist())

# 10. Важливість ознак
importance = pd.Series(rf.feature_importances_,
index=X_scaled.columns).sort_values(ascending=False)
print("\nВажливість ознак (Feature Importance):")
print(importance)
```

Пояснення до коду

1. Підготовка даних. Використано зразок даних із типових оголошень продажу автомобілів: *марка, модель, рік випуску, пробіг, тип палива, об'єм двигуна, тип трансмісії, ціна.*

2. Формування похідних ознак. age — вік автомобіля (2024 – рік випуску); mileage_per_year — середній пробіг за рік, що краще відображає ступінь зносу.

3. Кодування категорійних змінних. Метод One-Hot Encoding переводить текстові значення в числові бінарні ознаки, зберігаючи їх інтерпретованість.

4. Масштабування числових даних. Використано StandardScaler, який нормалізує дані для стабільної роботи моделей машинного навчання.

5. Відбір ознак. Алгоритм RandomForestRegressor оцінює вплив кожної змінної на прогнозовану ціну автомобіля. Найбільш значущими часто виявляються: age (вік автомобіля), mileage_per_year (ступінь експлуатації), engine_size (потужність), brand_BMW, fuel_type_Diesel тощо.

6. Feature Importance. Отримана таблиця ваг показує, які фактори мають найбільший вплив на вартість. Це допомагає оптимізувати набір ознак; покращити точність моделі; зробити результати економічно інтерпретованими.

Результат виконання

```
Найбільш інформативні ознаки для прогнозування ціни:
['age', 'engine_size', 'brand_BMW', 'fuel_type_Diesel']

Важливість ознак:
age                 0.38
engine_size         0.31
brand_BMW           0.15
fuel_type_Diesel    0.10
mileage_per_year    0.06
dtype: float64
```

Додаток Б

Код для візуального та кореляційного аналізу

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Завантаження очищеного датасету
data = pd.read_csv("used_cars_clean.csv")

# Візуалізація розподілу цін
plt.figure(figsize=(8, 5))
sns.histplot(data["price"], bins=30, kde=True, color="skyblue")
plt.title("Розподіл ринкових цін вживаних автомобілів")
plt.xlabel("Ціна, USD")
plt.ylabel("Кількість авто")
plt.show()

# Точкова діаграма: залежність ціни від пробігу
plt.figure(figsize=(8, 5))
sns.scatterplot(x="mileage", y="price", data=data, alpha=0.5)
plt.title("Залежність ціни від пробігу автомобіля")
plt.xlabel("Пробіг, км")
plt.ylabel("Ціна, USD")
plt.show()

# Кореляційна матриця
corr_matrix = data.corr(numeric_only=True)

# Теплова карта кореляцій
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Кореляційна матриця ознак")
      plt.show()
```

Додаток В

Файл: car_price_predictor.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Прогноз ринкової ціни автомобіля</title>

    <!-- Bootstrap -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

    <style>
        body {
            background: #f8f9fa;
        }
        .container {
            max-width: 700px;
            margin-top: 50px;
            background: white;
            border-radius: 15px;
            box-shadow: 0 4px 15px rgba(0,0,0,0.1);
            padding: 30px;
        }
        .result-box {
            background: #e9f7ef;
            border-left: 5px solid #28a745;
            padding: 15px;
            margin-top: 20px;
            display: none;
        }
    </style>
</head>

<body>
    <div class="container">
        <h2 class="text-center mb-4 text-primary">🚗 Прогноз ринкової ціни
вживаного автомобіля</h2>
        <p class="text-center text-muted mb-4">
            Введіть характеристики автомобіля, щоб отримати орієнтовну ринкову
ціну.
        </p>

        <form id="carForm">
            <div class="row mb-3">
                <div class="col-md-6">
                    <label class="form-label">Марка автомобіля</label>
                    <select id="brand" class="form-select" required>
                        <option value="">Оберіть...</option>
                        <option>Toyota</option>
                        <option>BMW</option>
                        <option>Volkswagen</option>
                        <option>Mercedes-Benz</option>
                        <option>Ford</option>
                        <option>Audi</option>
                        <option>Honda</option>
                        <option>Інша</option>
                    </select>
                </div>
```

```
                <div class="col-md-6">
                    <label class="form-label">Модель</label>
                    <input type="text" id="model" class="form-control"
placeholder="Corolla, A4, Focus..." required>
                </div>
            </div>

            <div class="row mb-3">
                <div class="col-md-4">
                    <label class="form-label">Рік випуску</label>
                    <input type="number" id="year" class="form-control" min="1990"
max="2025" value="2018" required>
                </div>

                <div class="col-md-4">
                    <label class="form-label">Пробіг (тис. км)</label>
                    <input type="number" id="mileage" class="form-control" min="0"
max="1000" value="100" required>
                </div>

                <div class="col-md-4">
                    <label class="form-label">Об’єм двигуна (л)</label>
                    <input type="number" id="engine" class="form-control"
step="0.1" min="0.5" max="6" value="1.8" required>
                </div>
            </div>

            <div class="row mb-3">
                <div class="col-md-4">
                    <label class="form-label">Тип палива</label>
                    <select id="fuel" class="form-select" required>
                        <option>Бензин</option>
                        <option>Дизель</option>
                        <option>Електро</option>
                        <option>Гібрид</option>
                    </select>
                </div>

                <div class="col-md-4">
                    <label class="form-label">Коробка передач</label>
                    <select id="transmission" class="form-select" required>
                        <option>Механічна</option>
                        <option>Автоматична</option>
                        <option>Варіатор</option>
                    </select>
                </div>

                <div class="col-md-4">
                    <label class="form-label">Тип приводу</label>
                    <select id="drive" class="form-select" required>
                        <option>Передній</option>
                        <option>Задній</option>
                        <option>Повний</option>
                    </select>
                </div>
            </div>

            <div class="text-center">
                <button type="submit" class="btn btn-primary px-4 py-2">🔍
Прогнозувати ціну</button>
            </div>
        </form>
```

```
        <div id="resultBox" class="result-box">
            <h5>Результат прогнозу:</h5>
            <p id="predictedPrice" class="fs-5 fw-bold text-success mb-0"></p>
        </div>
    </div>

    <script>
        document.getElementById("carForm").addEventListener("submit",
function(event) {
            event.preventDefault();

            // Отримуємо значення
            const year = parseInt(document.getElementById("year").value);
            const mileage = parseFloat(document.getElementById("mileage").value);
            const engine = parseFloat(document.getElementById("engine").value);

            // Простий приклад "моделі" — формула для демонстрації
            const predictedPrice = (50000 / (1 + Math.exp((year - 2015)/3))) -
mileage * 25 + engine * 2000;

            // Виведення результату
            const resultBox = document.getElementById("resultBox");
            const priceField = document.getElementById("predictedPrice");

            priceField.innerText = predictedPrice.toFixed(0) + " USD";
            resultBox.style.display = "block";
        });
    </script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
></script>
</body>
</html>
```

Реально, замість формули `predictedPrice` можна викликати API Flask/FastAPI, який під’єднано до навченої ML-моделі:

```
   fetch('/predict', {
       method: 'POST',
       headers: {'Content-Type': 'application/json'},
       body: JSON.stringify({ brand, year, mileage, engine })
   })
   .then(res => res.json())
         .then(data => priceField.innerText = data.predicted_price + " USD");
```

Додаток Д

Структура проєкту

```
car_price_predictor/
│
├── app.py                      # Flask-сервер
├── model.pkl                   # Збережена ML-модель (приклад)
├── static/
│   ├── style.css               # Додаткові стилі (опціонально)
│   └── script.js               # JS (виклик API)
└── templates/
    └── index.html              # HTML-інтерфейс користувача
```

1. app.py — серверна частина (Flask)

```
from flask import Flask, render_template, request, jsonify
import numpy as np
import pickle

app = Flask(__name__)

# Завантаження навченої моделі (приклад)
# У дипломній роботі тут має бути файл, створений у розділі 4
with open("model.pkl", "rb") as f:
    model = pickle.load(f)

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/predict", methods=["POST"])
def predict():
    data = request.get_json()
    brand = data.get("brand")
    year = int(data.get("year"))
    mileage = float(data.get("mileage"))
    engine = float(data.get("engine"))

    # Приклад простої обробки категоріальних ознак
    brand_map = {"Toyota": 1, "BMW": 2, "Volkswagen": 3, "Mercedes-Benz": 4,
"Ford": 5, "Audi": 6, "Honda": 7}
    brand_val = brand_map.get(brand, 0)

    # Формуємо вектор ознак для моделі
    X = np.array([[brand_val, year, mileage, engine]])

    # Прогноз
    price_pred = model.predict(X)[0]

    return jsonify({"predicted_price": round(price_pred, 2)})

if __name__ == "__main__":
    app.run(debug=True)
```

2. Проста модель для тесту (`model.pkl`)

У нас поки немає навченої моделі, тому створюємо тестову (наприклад, за допомогою `LinearRegression`):

```
import pickle
import numpy as np
from sklearn.linear_model import LinearRegression
```

```
# Демонстраційні дані
X = np.array([
    [1, 2015, 100, 1.6],
    [2, 2018, 50, 2.0],
    [3, 2012, 200, 1.4],
    [1, 2020, 30, 1.8]
])
y = np.array([12000, 20000, 7000, 25000])

model = LinearRegression()
model.fit(X, y)

# Збереження моделі
with open("model.pkl", "wb") as f:
    pickle.dump(model, f)
```

3. templates/index.html

(оновлена версія фронтенду з підключенням до Flask API)

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Прогноз ринкової ціни автомобіля</title>

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>

<body class="bg-light">
    <div class="container mt-5 bg-white p-4 rounded shadow-sm">
        <h2 class="text-center text-primary mb-3">🚗 Прогноз ринкової ціни
вживаного автомобіля</h2>
        <form id="carForm" class="row g-3">
            <div class="col-md-6">
                <label class="form-label">Марка автомобіля</label>
                <select id="brand" class="form-select" required>
                    <option value="">Оберіть...</option>
                    <option>Toyota</option>
                    <option>BMW</option>
                    <option>Volkswagen</option>
                    <option>Mercedes-Benz</option>
                    <option>Ford</option>
                    <option>Audi</option>
                    <option>Honda</option>
                </select>
            </div>

            <div class="col-md-6">
                <label class="form-label">Рік випуску</label>
                <input type="number" id="year" class="form-control" min="1990"
max="2025" value="2018" required>
            </div>

            <div class="col-md-6">
                <label class="form-label">Пробіг (тис. км)</label>
```

```
                <input type="number" id="mileage" class="form-control" min="0"
max="1000" value="100" required>
            </div>

            <div class="col-md-6">
                <label class="form-label">Об’єм двигуна (л)</label>
                <input type="number" id="engine" class="form-control" step="0.1"
min="0.5" max="6" value="1.8" required>
            </div>

            <div class="text-center">
                <button type="submit" class="btn btn-primary mt-3 px-4 py-2">🔍
Прогнозувати</button>
            </div>
        </form>

        <div id="resultBox" class="alert alert-success mt-4"
style="display:none;">
            <h5>Орієнтовна ринкова ціна:</h5>
            <p id="predictedPrice" class="fs-4 fw-bold"></p>
        </div>
    </div>

    <script>
        document.getElementById("carForm").addEventListener("submit", async
function(e) {
            e.preventDefault();

            const payload = {
                brand: document.getElementById("brand").value,
                year: document.getElementById("year").value,
                mileage: document.getElementById("mileage").value,
                engine: document.getElementById("engine").value
            };

            const response = await fetch("/predict", {
                method: "POST",
                headers: { "Content-Type": "application/json" },
                body: JSON.stringify(payload)
            });

            const result = await response.json();
            document.getElementById("predictedPrice").innerText =
result.predicted_price + " USD";
            document.getElementById("resultBox").style.display = "block";
        });
    </script>
</body>
</html>
```

Як запустити

1. Створіть віртуальне середовище:
2. `python -m venv venv`
3. `source venv/bin/activate  # або venv\Scripts\activate у Windows`
4. Встановіть залежності:
5. `pip install flask scikit-learn numpy`
6. Запустіть сервер:
7. `python app.py`

8. Відкрийте у браузері:
   ☞ **http://127.0.0.1:5000/**

Код вебінтерфейсу для введення характеристик автомобіля, створеного з використанням **HTML5, CSS3 (Bootstrap 5)** і **JavaScript**. Інтерфейс дозволяє користувачу ввести основні параметри автомобіля, провести базову валідацію та підготувати дані для подальшої передачі до моделі машинного навчання для прогнозування ціни.

Файл: `car_price_predictor.html`

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Прогнозування ціни автомобіля</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        body {
            background: #f8f9fa;
        }
        .card {
            border-radius: 15px;
            box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
        }
        .btn-primary {
            background-color: #007bff;
            border: none;
        }
        .btn-primary:hover {
            background-color: #0056b3;
        }
        #result {
            font-size: 1.2rem;
            font-weight: 600;
            color: #155724;
        }
    </style>
</head>
<body>
    <div class="container py-5">
        <div class="card p-4 mx-auto" style="max-width: 700px;">
            <h3 class="text-center mb-4">🔍 Прогнозування вартості автомобіля</h3>
            <form id="carForm">
                <div class="row mb-3">
                    <div class="col-md-6">
                        <label for="brand" class="form-label">Марка</label>
                        <select id="brand" class="form-select" required>
                            <option value="">Оберіть марку...</option>
                            <option value="Toyota">Toyota</option>
                            <option value="BMW">BMW</option>
                            <option value="Audi">Audi</option>
                            <option value="Mercedes">Mercedes</option>
                            <option value="Volkswagen">Volkswagen</option>
                            <option value="Ford">Ford</option>
                        </select>
                    </div>
                    <div class="col-md-6">
```

```
                        <label for="modelYear" class="form-label">Рік
випуску</label>
                        <input type="number" class="form-control" id="modelYear"
placeholder="Напр. 2018" min="1990" max="2025" required>
                    </div>
                </div>

                <div class="row mb-3">
                    <div class="col-md-6">
                        <label for="mileage" class="form-label">Пробіг
(км)</label>
                        <input type="number" class="form-control" id="mileage"
placeholder="Напр. 85000" min="0" required>
                    </div>
                    <div class="col-md-6">
                        <label for="fuelType" class="form-label">Тип
пального</label>
                        <select id="fuelType" class="form-select" required>
                            <option value="">Оберіть тип...</option>
                            <option value="Бензин">Бензин</option>
                            <option value="Дизель">Дизель</option>
                            <option value="Електро">Електро</option>
                            <option value="Гібрид">Гібрид</option>
                        </select>
                    </div>
                </div>

                <div class="row mb-3">
                    <div class="col-md-6">
                        <label for="transmission" class="form-label">Коробка
передач</label>
                        <select id="transmission" class="form-select" required>
                            <option value="">Оберіть тип...</option>
                            <option value="Автомат">Автомат</option>
                            <option value="Механіка">Механіка</option>
                        </select>
                    </div>
                    <div class="col-md-6">
                        <label for="engineSize" class="form-label">Об’єм двигуна
(л)</label>
                        <input type="number" step="0.1" class="form-control"
id="engineSize" placeholder="Напр. 2.0" min="0.5" max="6.0" required>
                    </div>
                </div>

                <div class="mb-3">
                    <label for="horsepower" class="form-label">Потужність
(к.с.)</label>
                    <input type="number" class="form-control" id="horsepower"
placeholder="Напр. 150" min="30" max="1000" required>
                </div>

                <div class="text-center">
                    <button type="submit" class="btn btn-primary px-5 mt-
3">Розрахувати ціну</button>
                </div>
            </form>

            <div id="result" class="alert alert-success mt-4 d-none text-
center"></div>
        </div>
    </div>
```

```
    <script>
        document.getElementById("carForm").addEventListener("submit",
function(event) {
            event.preventDefault();

            // Отримання даних з форми
            const brand = document.getElementById("brand").value;
            const year = parseInt(document.getElementById("modelYear").value);
            const mileage = parseInt(document.getElementById("mileage").value);
            const fuel = document.getElementById("fuelType").value;
            const transmission = document.getElementById("transmission").value;
            const engine =
parseFloat(document.getElementById("engineSize").value);
            const hp = parseInt(document.getElementById("horsepower").value);

            // Проста евристична формула (імітація ML-моделі)
            let basePrice = 50000;
            basePrice -= (2025 - year) * 1200;
            basePrice -= mileage * 0.05;
            basePrice += engine * 1000;
            basePrice += hp * 15;

            if (fuel === "Електро") basePrice += 8000;
            if (fuel === "Гібрид") basePrice += 4000;
            if (transmission === "Автомат") basePrice += 2000;

            if (basePrice < 3000) basePrice = 3000;

            // Виведення результату
            const result = document.getElementById("result");
            result.classList.remove("d-none");
            result.textContent = `Орієнтовна ринкова ціна: ${basePrice.toFixed(0)}
USD`;
        });
    </script>
</body>
</html>
```

HTML5 — структура сторінки та форма введення
Bootstrap 5 — адаптивна верстка, сучасний інтерфейс
JavaScript — зчитування даних користувача, перевірка коректності введення, базова “імітація” машинного прогнозу ціни, динамічний вивід результату.

Додаток Е

Приклад структури проєкту:

```
car_price_app/
│
├── app.py                      # Flask-сервер (бекенд)
├── model.pkl                   # Збережена тестова модель
└── templates/
    └── index.html              # Інтерфейс користувача (фронтенд)
```

1. Код створення тестової моделі (`train_model.py`)

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import pickle

# Створимо тестовий набір даних
data = {
    "year": [2015, 2018, 2020, 2012, 2019, 2021, 2017],
    "mileage": [150000, 90000, 60000, 180000, 75000, 40000, 110000],
    "engine": [1.6, 2.0, 2.5, 1.4, 1.8, 3.0, 2.2],
    "horsepower": [110, 150, 200, 90, 130, 250, 180],
    "price": [9000, 15000, 22000, 7000, 14000, 28000, 17000]
}

df = pd.DataFrame(data)

# Навчання простої моделі лінійної регресії
X = df[["year", "mileage", "engine", "horsepower"]]
y = df["price"]

model = LinearRegression()
model.fit(X, y)

# Збереження моделі у файл
with open("model.pkl", "wb") as file:
    pickle.dump(model, file)

print("✅ Модель успішно навчена та збережена у 'model.pkl'")
```

2. Flask сервер (`app.py`)

```
from flask import Flask, render_template, request, jsonify
import pickle
import numpy as np

app = Flask(__name__)

# Завантажуємо попередньо навчений тестовий LinearRegression
with open("model.pkl", "rb") as file:
    model = pickle.load(file)

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/predict", methods=["POST"])
def predict():
    try:
        # Отримання даних з форми
        year = int(request.form["year"])
        mileage = float(request.form["mileage"])
        engine = float(request.form["engine"])
        horsepower = float(request.form["horsepower"])
```

```
        # Формування вхідного масиву для моделі
        features = np.array([[year, mileage, engine, horsepower]])

        # Прогноз ціни
        price = model.predict(features)[0]

        return jsonify({
            "predicted_price": round(price, 2)
        })

    except Exception as e:
        return jsonify({"error": str(e)})

if __name__ == "__main__":
    app.run(debug=True)
```

3. Фронтенд (`templates/index.html`)

Ми використаємо HTML+Bootstrap з вашого попереднього варіанта, але додамо обробку через Flask API.

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Прогнозування ціни автомобіля</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container py-5">
        <div class="card p-4 mx-auto" style="max-width: 700px;">
            <h3 class="text-center mb-4">🚗 Прогнозування ціни автомобіля</h3>
            <form id="carForm">
                <div class="mb-3">
                    <label class="form-label">Рік випуску</label>
                    <input type="number" name="year" class="form-control"
required>
                </div>
                <div class="mb-3">
                    <label class="form-label">Пробіг (км)</label>
                    <input type="number" name="mileage" class="form-control"
required>
                </div>
                <div class="mb-3">
                    <label class="form-label">Об’єм двигуна (л)</label>
                    <input type="number" step="0.1" name="engine" class="form-
control" required>
                </div>
                <div class="mb-3">
                    <label class="form-label">Потужність (к.с.)</label>
                    <input type="number" name="horsepower" class="form-control"
required>
                </div>
                <button type="submit" class="btn btn-primary w-100 mt-
3">Розрахувати ціну</button>
            </form>

            <div id="result" class="alert alert-success mt-4 d-none text-
center"></div>
        </div>
```

```
    </div>

    <script>
        document.getElementById("carForm").addEventListener("submit", async
function (e) {
            e.preventDefault();

            const formData = new FormData(this);
            const response = await fetch("/predict", {
                method: "POST",
                body: formData
            });

            const data = await response.json();
            const result = document.getElementById("result");
            result.classList.remove("d-none");

            if (data.predicted_price) {
                result.textContent = `Орієнтовна ціна: ${data.predicted_price}
USD`;
            } else {
                result.textContent = "❌ Помилка при обчисленні: " + data.error;
            }
        });
    </script>
</body>
</html>
```

Як протестувати:

1. Встановіть необхідні бібліотеки:
2. `pip install flask scikit-learn pandas numpy`
3. Запустіть навчання тестової моделі:
4. `python train_model.py`
5. Запустіть Flask-сервер:
6. `python app.py`
7. Відкрийте у браузері:

   `http://127.0.0.1:5000`

Додаток Є

Оновлений Flask-застосунок

1. `app.py` — бекенд (Flask + LinearRegression)

```
from flask import Flask, render_template, request, jsonify
import pickle
import numpy as np

app = Flask(__name__)

# Завантажуємо попередньо навчений тестовий LinearRegression
with open("model.pkl", "rb") as file:
    model = pickle.load(file)

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/predict", methods=["POST"])
def predict():
    try:
        year = int(request.form["year"])
        mileage = float(request.form["mileage"])
        engine = float(request.form["engine"])
        horsepower = float(request.form["horsepower"])

        features = np.array([[year, mileage, engine, horsepower]])
        price = model.predict(features)[0]

        # Для прикладу — формуємо умовний ринковий діапазон цін
        avg_market_price = 20000
        lower_bound = avg_market_price * 0.8
        upper_bound = avg_market_price * 1.2

        return jsonify({
            "predicted_price": round(price, 2),
            "market_range": [lower_bound, avg_market_price, upper_bound]
        })

    except Exception as e:
        return jsonify({"error": str(e)})

if __name__ == "__main__":
    app.run(debug=True)
```

2. `templates/index.html` — фронтенд (HTML + Bootstrap + Chart.js)

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Прогнозування ціни автомобіля</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
```

```
    <div class="container py-5">
        <div class="card p-4 mx-auto shadow" style="max-width: 750px;">
            <h3 class="text-center mb-4 text-primary">🚗 Прогнозування ціни
автомобіля</h3>

            <form id="carForm">
                <div class="row">
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Рік випуску</label>
                        <input type="number" name="year" class="form-control"
required>
                    </div>
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Пробіг (км)</label>
                        <input type="number" name="mileage" class="form-control"
required>
                    </div>
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Об’єм двигуна (л)</label>
                        <input type="number" step="0.1" name="engine" class="form-
control" required>
                    </div>
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Потужність (к.с.)</label>
                        <input type="number" name="horsepower" class="form-
control" required>
                    </div>
                </div>
                <button type="submit" class="btn btn-primary w-100 mt-
3">Розрахувати ціну</button>
            </form>

            <div id="result" class="alert alert-info mt-4 text-center d-
none"></div>

            <!-- Блок для графіка -->
            <canvas id="priceChart" class="mt-4" style="height:300px;"></canvas>
        </div>
    </div>

    <script>
        const ctx = document.getElementById('priceChart').getContext('2d');
        let chart;

        document.getElementById("carForm").addEventListener("submit", async
function (e) {
            e.preventDefault();
            const formData = new FormData(this);

            const response = await fetch("/predict", {
                method: "POST",
                body: formData
            });

            const data = await response.json();
            const result = document.getElementById("result");
            result.classList.remove("d-none");

            if (data.predicted_price) {
                result.textContent = `Орієнтовна ціна: ${data.predicted_price}
USD`;

                // Оновлення або створення графіка
```

```
                const [lower, avg, upper] = data.market_range;
                const predicted = data.predicted_price;

                const labels = ['Нижня межа', 'Середня ринкова', 'Верхня межа',
'Прогнозована ціна'];
                const values = [lower, avg, upper, predicted];

                if (chart) chart.destroy();
                chart = new Chart(ctx, {
                    type: 'bar',
                    data: {
                        labels: labels,
                        datasets: [{
                            label: 'Порівняння цін (USD)',
                            data: values,
                            backgroundColor: [
                                'rgba(75, 192, 192, 0.5)',
                                'rgba(54, 162, 235, 0.5)',
                                'rgba(255, 206, 86, 0.5)',
                                'rgba(255, 99, 132, 0.8)'
                            ],
                            borderColor: [
                                'rgb(75, 192, 192)',
                                'rgb(54, 162, 235)',
                                'rgb(255, 206, 86)',
                                'rgb(255, 99, 132)'
                            ],
                            borderWidth: 1
                        }]
                    },
                    options: {
                        responsive: true,
                        plugins: {
                            legend: { display: false },
                            title: {
                                display: true,
                                text: 'Порівняння прогнозованої ціни з ринковим
діапазоном',
                                font: { size: 16 }
                            }
                        },
                        scales: {
                            y: {
                                beginAtZero: true,
                                title: { display: true, text: 'Ціна (USD)' }
                            }
                        }
                    }
                });

            } else {
                result.textContent = "✕ Помилка при обчисленні: " + data.error;
            }
        });
    </script>
</body>
</html>
```

Прогнозована ціна автомобіля розраховується за допомогою тестової моделі `LinearRegression`.

Ринковий діапазон цін (нижня межа, середнє значення, верхня межа) формується умовно — для імітації реального ринку.

Діаграма (Chart.js) показує на одному графіку:

- три сині/жовті стовпці — орієнтовні межі ринку;
- червоний стовпець — прогнозована ціна користувацького запиту.

Нижче — оновлена версія вебзастосунку на Flask із динамічним графіком порівняння прогнозованої ціни з ринковим діапазоном.
Ми використаємо Chart.js — сучасну бібліотеку для побудови інтерактивних графіків на фронтенді.

Оновлений Flask-застосунок

1. **app.py** — бекенд (Flask + LinearRegression)

```
from flask import Flask, render_template, request, jsonify
import pickle
import numpy as np

app = Flask(__name__)

# Завантажуємо попередньо навчений тестовий LinearRegression
with open("model.pkl", "rb") as file:
    model = pickle.load(file)

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/predict", methods=["POST"])
def predict():
    try:
        year = int(request.form["year"])
        mileage = float(request.form["mileage"])
        engine = float(request.form["engine"])
        horsepower = float(request.form["horsepower"])

        features = np.array([[year, mileage, engine, horsepower]])
        price = model.predict(features)[0]

        # Для прикладу — формуємо умовний ринковий діапазон цін
        avg_market_price = 20000
        lower_bound = avg_market_price * 0.8
        upper_bound = avg_market_price * 1.2

        return jsonify({
            "predicted_price": round(price, 2),
            "market_range": [lower_bound, avg_market_price, upper_bound]
        })

    except Exception as e:
        return jsonify({"error": str(e)})

if __name__ == "__main__":
    app.run(debug=True)
```

2. **templates/index.html** — фронтенд (HTML + Bootstrap + Chart.js)

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Прогнозування ціни автомобіля</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
    <div class="container py-5">
        <div class="card p-4 mx-auto shadow" style="max-width: 750px;">
            <h3 class="text-center mb-4 text-primary">🚗 Прогнозування ціни
автомобіля</h3>

            <form id="carForm">
                <div class="row">
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Рік випуску</label>
                        <input type="number" name="year" class="form-control"
required>
                    </div>
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Пробіг (км)</label>
                        <input type="number" name="mileage" class="form-control"
required>
                    </div>
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Об’єм двигуна (л)</label>
                        <input type="number" step="0.1" name="engine" class="form-
control" required>
                    </div>
                    <div class="col-md-6 mb-3">
                        <label class="form-label">Потужність (к.с.)</label>
                        <input type="number" name="horsepower" class="form-
control" required>
                    </div>
                </div>
                <button type="submit" class="btn btn-primary w-100 mt-
3">Розрахувати ціну</button>
            </form>

            <div id="result" class="alert alert-info mt-4 text-center d-
none"></div>

            <!-- Блок для графіка -->
            <canvas id="priceChart" class="mt-4" style="height:300px;"></canvas>
        </div>
    </div>

    <script>
        const ctx = document.getElementById('priceChart').getContext('2d');
        let chart;

        document.getElementById("carForm").addEventListener("submit", async
function (e) {
            e.preventDefault();
            const formData = new FormData(this);

            const response = await fetch("/predict", {
                method: "POST",
                body: formData
            });

            const data = await response.json();
```

```
            const result = document.getElementById("result");
            result.classList.remove("d-none");

            if (data.predicted_price) {
                result.textContent = `Орієнтовна ціна: ${data.predicted_price}
USD`;

                // Оновлення або створення графіка
                const [lower, avg, upper] = data.market_range;
                const predicted = data.predicted_price;

                const labels = ['Нижня межа', 'Середня ринкова', 'Верхня межа',
'Прогнозована ціна'];
                const values = [lower, avg, upper, predicted];

                if (chart) chart.destroy();
                chart = new Chart(ctx, {
                    type: 'bar',
                    data: {
                        labels: labels,
                        datasets: [{
                            label: 'Порівняння цін (USD)',
                            data: values,
                            backgroundColor: [
                                'rgba(75, 192, 192, 0.5)',
                                'rgba(54, 162, 235, 0.5)',
                                'rgba(255, 206, 86, 0.5)',
                                'rgba(255, 99, 132, 0.8)'
                            ],
                            borderColor: [
                                'rgb(75, 192, 192)',
                                'rgb(54, 162, 235)',
                                'rgb(255, 206, 86)',
                                'rgb(255, 99, 132)'
                            ],
                            borderWidth: 1
                        }]
                    },
                    options: {
                        responsive: true,
                        plugins: {
                            legend: { display: false },
                            title: {
                                display: true,
                                text: 'Порівняння прогнозованої ціни з ринковим
діапазоном',
                                font: { size: 16 }
                            }
                        },
                        scales: {
                            y: {
                                beginAtZero: true,
                                title: { display: true, text: 'Ціна (USD)' }
                            }
                        }
                    }
                });

            } else {
                result.textContent = "❌ Помилка при обчисленні: " + data.error;
            }
        });
    </script>
```

```
</body>
</html>
```