МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

ПРОГНОЗУВАННЯ ЦІН НОУТБУКІВ ІЗ ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ – 608м.21810802

Здобувачка _____ Ю. А. Андреєва
підпис

«__» _____ 2024 р.

Керівник PhD, ст. викладач кафедри ІПЗ _____ К. О. Антіпова
підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко

«_____» _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

Видано студенту групи 608м факультету комп'ютерних наук

Андреєвій Юлії Андріївні .

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

«ПРОГНОЗУВАННЯ ЦІН НОУТБУКІВ ІЗ ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ» .

Затверджена наказом по ЧНУ від «10» листопада 2023 р. № 234

2. Строк представлення кваліфікаційної роботи «____» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Вхідні дані до роботи − датасет. Результат – функціонуючий вебзастосунок прогнозування цін ноутбуків.

4. Перелік питань, що підлягають розробці

- аналіз методологій розробки програмного забезпечення, тобто дослідження предметної області;
- огляд існуючих вебзастосунків прогнозування цін;

- розробка вимог до системи на основі інформації про предметну область та існуючі аналоги;
- розробка алгоритму вебзастосунку;
- проєктування вебзастосунку прогнозування цін;
- програмна реалізація, тестування та відлагодження вебзастосунку для прогнозування цін.

5. Перелік графічних матеріалів

Презентація.

Керівник роботи ст. викладач кафедри ІПЗ Антіпова Катерина Олександрівна.
(посада, прізвище, ім'я, по батькові)

_____
(підпис)

Завдання прийнято до виконання

Андреєва Юлія Андріївна.
(прізвище, ім'я, по батькові студента)

_____
(підпис)

Дата видачі завдання «____» _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: «Прогнозування цін ноутбуків із використанням методів машинного навчання»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	10.10.2023	11.10.2013	Виконано
2.	Огляд літератури за темою роботи	12.10.2023	15.10.2023	Виконано
3.	Складання календарного плану КРБ	16.10.2023	18.10.2023	Виконано
4.	Аналіз предметної області	19.10.2023	27.10.2023	Виконано
5.	Розробка проєктних рішень	28.10.2023	02.11.2023	Виконано
6.	Моделювання та конструювання ПЗ	03.11.2023	06.11.2023	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	07.11.2023	25.12.2023	Виконано
8.	Відгук керівника КРБ	16.02.2024	16.02.2024	Виконано
9.	Оформлення КРБ та презентації	18.01.2024	27.01.2024	Виконано
10.	Попередній захист	08.02.2024	08.02.2024	Виконано
11.	Рецензування	18.02.2024	19.02.2024	Виконано
12.	Завершення оформлення КРБ та презентації	09.02.2024	15.02.2024	Виконано
13.	Захист кваліфікаційної роботи	27.02.2024	27.02.2024	Виконано

Розробив студент Андреєва Юлія Андріївна . _____.
(прізвище, ім'я, по батькові студента) (*підпис*)

«____» _____ 2023 р.

Керівник роботи PhD, ст. викладач кафедри ІПЗ Антіпова Катерина Олександрівна .
(посада, прізвище, ім'я, по батькові) (*підпис*)

«____» _____ 2023

АНОТАЦІЯ

до кваліфікаційної роботи магістра

«Прогнозування цін ноутбуків із використанням методів машинного навчання»

Студент 608м гр.: Андреєва Юлія Андріївна

Керівник: PhD, ст. викладач кафедри ІПЗ Антіпова К. О.

Актуальність теми зумовлена зумовлена кількома важливими факторами:

- Змінами на ринку. Ринок ноутбуків є вкрай динамічним і піддається постійним змінам в технологічних та маркетингових трендах. В такому середовищі точні прогнози щодо цін стають важливими для виробників, роздрібних та оптових продавців, а також споживачів.
- Економічними вигодами. Точні прогнози цін дозволяють оптимізувати стратегії ціноутворення та запасів, що може призвести до збільшення прибутку та зменшення витрат.
- Популярністю онлайн-торгівлі. Зростаюча популярність онлайн-торгівлі робить цінову конкуренцію більш інтенсивною, і точні прогнози цін можуть допомогти компаніям зберігати конкурентну перевагу.
- Залученням споживачів. Покупці користуються веб-сайтами та додатками для порівняння цін, тож точні прогнози допомагають їм приймати обдумані рішення.
- Аналізом великих даних. Збільшується важливість аналізу великих обсягів даних у бізнесі та різних галузях. Машинне навчання є інструментом, що допомагає виробникам та роздрібним продавцям аналізувати та прогнозувати ціни великої кількості товарів.

Об’єктом дослідження є процеси прогнозування цін ноутбуків.

Предметом дослідження є методи регресії для обробки великих обсягів даних.

Метою дослідження є покращення точності прогнозування цін ноутбуків за рахунок розробки застосунку для аналізу великого набору даних із використанням методів машинного навчання.

У першому розділі представлений опис методів регресії, наведений алгоритм навчання моделі, розроблена специфікація вимог до програмного забезпечення. У другому розділі наведено опис предметної області, опис набору вхідних даних та ознак, первинний аналіз даних, їхня передобробка та візуалізація. У третьому розділі здійснено проєктування та моделювання системи, вибір технології та мови програмування, вибір компонентів програмного забезпечення. У четвертому розділі показано основні кроки реалізації застосунку. В останньому розділі розглянуто питання охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення.

В результаті виконання кваліфікаційної роботи магістра було реалізовано вебзастосунок прогнозування цін ноутбуків із використанням методів машинного навчання. Також були зроблені висновки щодо покращення точності прогнозування цін ноутбуків.

КРМ викладена на 90 сторінок, вона містить 5 розділів, 38 ілюстрацій, 2 таблиць, 19 джерел в переліку посилань

Ключові слова: машинне навчання, вебзастосунок, прогнозування, обробка даних, аналіз даних, Python, Django.

ABSTRACT

of the Master's Thesis

"Prediction of laptop prices using machine learning methods"

Student of group 608m: Andreieva Yuliia Andriivna

Supervisor: PhD, St. Lecturer of the Department of SE Antipova K.O.

The relevance of the topic is determined by several important factors:

- Changes in the market. The laptop market is extremely dynamic and subject to constant changes in technological and marketing trends. In such an environment, accurate price forecasts become important for manufacturers, retailers and wholesalers, as well as consumers.
- Economic benefits. Accurate price forecasts enable optimization of pricing and inventory strategies, which can lead to increased profits and reduced costs.
- Popularity of online trade. The growing popularity of online shopping makes price competition more intense, and accurate price forecasts can help companies maintain a competitive edge.
- Involvement of consumers. Shoppers use websites and apps to compare prices, so accurate forecasts help them make informed decisions.
- Big data analysis. The importance of analyzing large volumes of data in business and various industries is increasing. Machine learning is a tool that helps manufacturers and retailers analyze and predict the prices of a large number of products.

The object of the study is the process of forecasting laptop prices.

The subject of research is regression methods for processing large volumes of data.

The aim of the study is to improve the accuracy of laptop price forecasting by developing an application for analyzing a large data set using machine learning methods.

The first section presents a description of regression methods, a model learning algorithm, and developed a specification of software requirements. The second section provides a description of the subject area, a description of the set of input data and features,

primary data analysis, their refinement and visualization. In the third section, the design and modeling of the system, the choice of technology and programming language, and the selection of software components are carried out. The fourth section shows the main steps of application implementation. The last section deals with occupational health and safety issues that are directly related to the activities of a software developer.

As a result of completing the master's qualification work, a web application for predicting laptop prices using machine learning methods was implemented. Conclusions were also made regarding the improvement of the accuracy of forecasting laptop prices

The KRM is set out on 90 pages, it contains 5 chapters, 38 illustrations, 2 tables, 19 sources in the list of references

Keywords: machine learning, web application, forecasting, data processing, data analysis, Python, Django.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

КРМ – кваліфікаційна робота магістра

ПЗ – програмне забезпечення

ІПЗ – інженерія програмного забезпечення

ПІБ – прізвище, ім’я, по батькові

ЧНУ – Чорноморський національний університет

СУБД – системи управління базами даних

SGD Regression – Stochastic Gradient Descent Regression

KNN Regressor – k-Nearest Neighbors Regressor

SVR – Support Vector Regression

SVM – Support Vector Machine

MLRegressor – Multi-Layer Perceptron Regressor

ADABoost Regressor – Adaptive Boosting Regressor

XGBoost – eXtreme Gradient Boosting

MAE – Mean Absolute Error

RMSE – Root Mean Squared Error

MSE – Mean Squared Error

ГБ – гігабайт

REST – Representational State Transfer

HTTPS – Hypertext Transfer Protocol Secure

SQL – Structured Query Language

RAM – Random Access Memory

OS – Operating System

SSD – Solid State Drive

HDD – Hard Disk Drive

ВСТУП

Сучасний ринок ноутбуків постійно змінюється, пропозиція та попит зростають, що робить точне прогнозування цін на цифрові пристрої дуже важливим завданням для бізнесу та споживачів. В контексті стрімких змін у технологічних трендах та маркетингових стратегіях виробників, прогнози цін на ноутбуки можуть бути неточними, що впливає на прийняття стратегічних рішень та планування закупівель. Для покращення точності прогнозування цін на ноутбуки та забезпечення більш ефективного ринкового аналізу в цьому динамічному оточенні доцільно використовувати методи машинного навчання.

Машинне навчання стає все більш потужним і важливим інструментом для аналізу великих обсягів даних та прогнозування. Це дозволяє враховувати багато чинників, які можуть впливати на ціни ноутбуків, і розробляти моделі, які можуть адаптуватися до змін на ринку. Зростання обсягів даних про ноутбуки та їх характеристики ставить виклики перед аналітиками та дослідниками. Використання машинного навчання дозволяє ефективно аналізувати великі набори даних та витягувати корисну інформацію з них. Тому розробка моделі машинного навчання для аналізу відомостей про ноутбуки, а також прогнозування цін є актуальною та важливою задачею.

Отже, **актуальність** теми зумовлена зумовлена кількома важливими факторами:

– Змінами на ринку. Ринок ноутбуків є вкрай динамічним і піддається постійним змінам в технологічних та маркетингових трендах. В такому середовищі точні прогнози щодо цін стають важливими для виробників, роздрібних та оптових продавців, а також споживачів.

- Економічними вигодами. Точні прогнози цін дозволяють оптимізувати стратегії ціноутворення та запасів, що може призвести до збільшення прибутку та зменшення витрат.
- Популярністю онлайн-торгівлі. Зростаюча популярність онлайн-торгівлі робить цінову конкуренцію більш інтенсивною, і точні прогнози цін можуть допомогти компаніям зберігати конкурентну перевагу.
- Залученням споживачів. Покупці користуються веб-сайтами та додатками для порівняння цін, тож точні прогнози допомагають їм приймати обдумані рішення.
- Аналізом великих даних. Збільшується важливість аналізу великих обсягів даних у бізнесі та різних галузях. Машинне навчання є інструментом, що допомагає виробникам та роздрібним продавцям аналізувати та прогнозувати ціни великої кількості товарів.

Об’єктом кваліфікаційної роботи є процеси прогнозування цін ноутбуків.

Предметом кваліфікаційної роботи є методи регресії для обробки великих обсягів даних.

Метою кваліфікаційної роботи є покращення точності прогнозування цін ноутбуків за рахунок розробки застосунку для аналізу великого набору даних із використанням методів машинного навчання.

Для досягнення поставленої мети необхідно виконати **наступні завдання**:

1) Зібрати дані для аналізу.
2) Провести описовий аналіз даних.
3) Провести попередню обробку та підготовку даних.
4) Побудувати модель машинного навчання та провести її навчання на навчальному наборі даних.

5) Оцінити точність моделі на тестовому наборі даних, використовуючи метрики якості.
6) Виконати аналіз результатів, отриманих стосовно вхідного датасету. Вибрати алгоритм машинного навчання, який найкраще підходить для даної задачі.
7) Провести моделювання застосунку для аналізу цін ноутбуків.
8) Розробити застосунок для аналізу цін ноутбуків.
9) Провести тестування розробленого застосунку та оцінити його ефективність та зручність для користувачів.
10) Оформлення звіту.

1 АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ

1.1 Навчання з учителем

Машинне навчання (Machine Learning) є галуззю штучного інтелекту, яка досліджує методи, алгоритми та моделі, що дозволяють комп'ютерам навчатися і покращувати свою продуктивність на основі даних, без явного програмування. Основні види машинного навчання включають:

Навчання з учителем (Supervised Learning) - це один із типів завдань машинного навчання, в якому модель навчається на основі вхідних даних та відповідних вихідних міток або цільових значень. У цьому типі навчання модель "вчителя" надаються правильні відповіді на певні вхідні дані, і завдання моделі полягає в тому, щоб навчитися прогнозувати правильні відповіді для нових вхідних даних.

Основні характеристики навчання з учителем включають:

Вхідні та вихідні дані. На вхід моделі надаються вхідні дані (ознаки або фічі) і відповідні вихідні мітки або цільові значення.

Цільова функція (Target Function). Модель навчається знаходити функцію, яка відображає вхідні дані на вихідні мітки. Ця функція називається цільовою функцією.

Навчальний набір даних. Для навчання моделі використовують навчальний набір даних, який містить пари вхідних даних та відповідних вихідних міток.

Алгоритми навчання. Використовуються різні алгоритми машинного навчання для навчання моделі, такі як лінійна регресія, дерева рішень, випадковий ліс, нейронні мережі та інші.

Оцінка моделі. Після навчання моделі її ефективність оцінюється на тестовому наборі даних для визначення її точності та загальної якості.

Прикладами задач навчання з учителем є:

Класифікація. Модель навчається визначати категорію або клас для вхідних даних, наприклад, розпізнавання зображень (детектування об'єктів, розпізнавання облич), визначення категорії електронної пошти (спам чи не спам) і т. д.

Регресія. Модель навчається прогнозувати числове значення для вхідних даних, наприклад, прогнозування цін на нерухомість, прибутку компанії, температури тощо.

Машинний переклад. Модель навчається перекладати тексти з однієї мови на іншу на основі навчального корпусу текстів.

Навчання з учителем є одним із найпоширеніших підходів в машинному навчанні і знаходить застосування в багатьох сферах, включаючи комп'ютерне зорове сприйняття, обробку природної мови, аналітику даних, фінанси та багато інших.

1.2 Алгоритм навчання моделі

Після попередньої обробки даних та візуального аналізу можна перейти до навчання моделі. Навчання моделі включає підготовку даних для використання в алгоритмах машинного навчання, вибір моделі та навчання її на тренувальних даних, а також оцінку її продуктивності. Основні кроки навчання моделі:

1) Розбиття даних. Дані зазвичай розбиваються на тренувальний набір, валідаційний набір та тестовий набір. Тренувальний набір використовується для навчання моделі, валідаційний набір використовується для налаштування гіперпараметрів моделі та оцінки її продуктивності, а тестовий набір використовується для остаточної оцінки моделі.
2) Вибір моделі. Вибирається підходящий алгоритм або модель машинного навчання для вирішення конкретної задачі. Це може бути лінійна регресія, дерева рішень, випадковий ліс, градієнтний бустінг або інші моделі.
3) Навчання моделі. Використовуючи тренувальний набір даних, модель навчається знаходити залежності між вхідними змінними та цільовою

змінною. Це включає оптимізацію параметрів моделі, щоб мінімізувати помилку прогнозування.

4) Налаштування гіперпараметрів. Гіперпараметри моделі, такі як швидкість навчання, кількість дерев, регуляризаційні параметри та інші, налаштовуються на валідаційному наборі даних для досягнення найкращої продуктивності моделі.
5) Оцінка моделі. Після навчання моделі та налаштування гіперпараметрів потрібно оцінити її продуктивність. Це можна зробити за допомогою різних метрик оцінки, які вимірюють точність та ефективність моделі.

1.3 Регресійні методи

У даній кваліфікаційній роботі було використано кілька алгоритмів машинного навчання [2] для аналізу відомостей про ноутбуки та їх характеристики. Далі розглянемо кожен алгоритм детальніше.

Linear Regression

Linear Regression – це метод машинного навчання, який використовується для передбачення числових значень залежної змінної на основі декількох незалежних змінних. Лінійна регресія шукає лінійну залежність між вхідними змінними і вихідною змінною. Лінійна залежність може бути виражена за допомогою формули:

$$y = b0 + b1 * x1 + b2 * x2 + ... + bn * xn,$$

де y - вихідна змінна,

b0 - константа,

b1 - коефіцієнт для змінної x1,

b2 - коефіцієнт для змінної x2 і т. д.,

xn - значення n-ої змінної.

Ridge

Ridge regression – це техніка регуляризації, яка використовується в лінійній регресії для вирішення проблеми мультиколінеарності, яка виникає, коли прогностичні змінні сильно корельовані. Він додає штрафний термін до функції втрат моделі лінійної регресії, який називається членом регуляризації L2, який допомагає зменшити величину коефіцієнтів.

Член регуляризації L2 у регресії Ridge обчислюється як сума квадратів коефіцієнтів, помножених на гіперпараметр, який називається параметром регуляризації (альфа). Збільшуючи значення альфа, величина коефіцієнтів зменшується до нуля, зменшуючи їх дисперсію та мінімізуючи вплив мультиколінеарності.

Ridge regression може допомогти запобігти переобладнанню в моделях лінійної регресії та покращити їх ефективність узагальнення. Він балансує між правильною підгонкою навчальних даних і збереженням малих коефіцієнтів, що може призвести до кращих прогнозів щодо нових, невідомих даних.

Lasso

Lasso (Least Absolute Shrinkage and Selection Operator) – ще один метод регуляризації, що використовується в лінійній регресії. Подібно до регресії Ridge, Lasso також вирішує проблему мультиколінеарності та допомагає зменшити величину коефіцієнтів. Однак Lasso використовує інший термін штрафу, який називається терміном регулярізації L1.

Термін регуляризації L1 у Lasso обчислюється як сума абсолютних значень коефіцієнтів, помножених на гіперпараметр, який називається параметром регуляризації (альфа). Lasso заохочує розрідженість значень коефіцієнтів, тобто прагне встановити деякі коефіцієнти рівними нулю, ефективно виконуючи вибір

функцій. Це робить Lasso особливо корисним при роботі з великовимірними даними, де релевантною може бути лише підмножина функцій.

Lasso можна використовувати для виконання вибору змінних шляхом автоматичного визначення та виключення нерелевантних або зайвих функцій із моделі. Це може допомогти спростити модель шляхом видалення менш важливих предикторів, сприяючи кращій інтерпретації та потенційному покращенню ефективності прогнозування.

DecisionTreeRegressor

DecisionTreeRegressor – це алгоритм машинного навчання, який належить до сімейства моделей дерева рішень для задач регресії. Він використовується для побудови моделі дерева рішень, яка може передбачати безперервні числові значення.

У дереві рішень вхідні дані розбиваються на підмножини на основі значень ознак, і кожен розподіл визначається правилом прийняття рішень. Для завдань регресії регресор дерева рішень будує структуру дерева, де кожен внутрішній вузол представляє ознаку та критерій поділу, а кожен кінцевий вузол представляє прогнозоване значення.

Регресор дерева рішень працює шляхом рекурсивного поділу даних на основі вибраної функції та критерію поділу, доки не буде виконано умову зупинки. Критерій розподілу зазвичай вибирається для мінімізації дисперсії або середньоквадратичної помилки цільової змінної в межах кожної підмножини.

Після того, як регресор дерева рішень навчено, його можна використовувати для прогнозування нових, невидимих даних, обходячи структуру дерева та призначаючи прогнозоване значення, пов’язане з досягнутим листовим вузлом.

SGD Regression

Регресія SGD, також відома як стохастична градієнтна регресія – це алгоритм лінійної регресії, який використовує оптимізацію стохастичного градієнта, щоб мінімізувати функцію витрат і знайти оптимальні коефіцієнти регресії.

У регресії SGD навчальні дані відбираються випадковим чином невеликими партіями під час кожної ітерації, а градієнт функції витрат обчислюється для кожної партії. Потім параметри моделі оновлюються з використанням інформації про градієнт, що допомагає ітеративно мінімізувати функцію витрат і покращити відповідність моделі навчальним даним.

Регресія SGD особливо корисна для проблем великомасштабної регресії або під час роботи з масивами даних великого розміру, оскільки вона може ефективно обробляти велику кількість зразків і функцій. Він також є гнучким і може вміщувати різні типи функцій втрат, методи регулярізації та графіки швидкості навчання.

Щоб використовувати регресію SGD, потрібно вказати функцію втрат, термін регуляризації та швидкість навчання. Зазвичай використовувані функції втрат включають середню квадратичну помилку (MSE) і середню абсолютну помилку (MAE), тоді як терміни регуляризації, такі як L1 (Lasso) і L2 (Ridge), можуть використовуватися для контролю складності моделі та запобігання переобладнанню.

Загалом SGD Regression є популярним вибором для завдань регресії завдяки своїй ефективності та здатності обробляти великомасштабні набори даних.

KNN Regressor

Регресор KNN, також відомий як k-регресор найближчих сусідів – це непараметричний алгоритм регресії, який прогнозує цільову змінну шляхом усереднення значень її k найближчих сусідів у просторі ознак.

У регресії KNN алгоритм спочатку ідентифікує k найближчих сусідів даної точки даних на основі їх схожості ознак. Потім цільова змінна для точки даних

прогнозується шляхом взяття середнього (або середньозваженого) цільових значень цих k сусідів. Вибір k визначає розмір локальної околиці та впливає на плавність регресійної моделі.

Регресія KNN – це простий, але потужний алгоритм, який може фіксувати складні нелінійні зв’язки між ознаками та цільовою змінною. Це ледачий алгоритм навчання, що означає, що він явно не вивчає модель під час фази навчання. Натомість він зберігає весь навчальний набір даних у пам’яті для ефективного пошуку під час фази прогнозування.

Одним із важливих моментів під час використання регресії KNN є вибір метрики відстані, яка вимірює подібність між точками даних. Загальні показники відстані включають евклідову відстань і манхеттенську відстань. Крім того, часто рекомендується масштабування або нормалізація функцій, щоб переконатися, що всі функції однаково сприяють обчисленню відстані.

Регресія KNN підходить як для безперервних, так і для дискретних цільових змінних. Його часто використовують у випадках, коли основний зв’язок між ознаками та цільовою змінною є складним і нелегко охопити параметричними моделями. Однак він може не працювати належним чином, якщо набір даних має велику кількість функцій або коли в даних є викиди. Це також обчислювально дорого під час фази прогнозування, оскільки вимагає обчислення відстані до всіх екземплярів навчання.

Загалом регресія KNN — це гнучкий та інтуїтивно зрозумілий алгоритм для завдань регресії, і на його продуктивність може впливати вибір k і метрики відстані. Його часто використовують у випадках, коли основний зв’язок між ознаками та цільовою змінною є складним і нелегко охопити параметричними моделями.

Random Forest

Random Forest [12] - це алгоритм машинного навчання, який використовується для класифікації та регресії. Це потужний і широко використовуваний алгоритм, який

поєднує прогнози кількох дерев рішень для створення більш точних і надійних прогнозів.

У RandomForestRegressor створюється велика кількість дерев рішень, кожне з яких навчається на випадковій підмножині навчальних даних. Під час навчання кожне дерево вирощується шляхом випадкового вибору підмножини функцій у кожній точці розділення. Така випадковість допомагає зменшити переобладнання та збільшити різноманітність серед дерев.

Остаточний прогноз RandomForestRegressor отримується шляхом усереднення прогнозів усіх окремих дерев у ансамблі. Процес усереднення допомагає згладити прогнози окремого дерева та забезпечити більш стабільний і точний прогноз.

RandomForestRegressor має кілька переваг. Він може обробляти як безперервні, так і категоричні функції, не вимагаючи значної попередньої обробки даних. Він також стійкий до викидів і відсутніх значень у даних. Крім того, RandomForestRegressor може фіксувати складні нелінійні зв’язки між функціями та цільовою змінною.

Алгоритм надає важливі показники для оцінки ефективності моделі, такі як середня квадратична помилка (MSE) і R-квадрат. Ці показники допомагають оцінити точність і відповідність моделі.

Однак RandomForestRegressor також має деякі обмеження. Це може бути дорогим з точки зору обчислень, особливо для великих наборів даних і великої кількості дерев. Крім того, можливість інтерпретації моделі може бути обмеженою порівняно з простішими моделями, такими як лінійна регресія.

Загалом RandomForestRegressor — це універсальний і потужний алгоритм для завдань регресії. Він широко використовується в різних сферах, зокрема у фінансах, охороні здоров’я та роздрібній торгівлі, де потрібні точні прогнози.

ADABoost Regressor

ADABoostRegressor [6] — це комплексний алгоритм навчання, який поєднує кілька слабких учнів (регресійні моделі) для створення сильної регресійної моделі. Він належить до сімейства алгоритмів підвищення, які ітеративно навчають слабкі моделі та зосереджуються на зразках, які раніше були неправильно спрогнозовані, щоб покращити загальну продуктивність.

Алгоритм ADABoostRegressor працює шляхом послідовної підгонки серії слабких регресійних моделей до навчальних даних. У кожній ітерації алгоритм призначає вищі ваги зразкам, які раніше були неправильно спрогнозовані, що дозволяє наступним моделям більше зосереджуватися на цих складних зразках. Слабкими моделями зазвичай є неглибокі дерева рішень або прості лінійні регресори.

Під час процесу навчання алгоритм ADABoostRegressor призначає вагові коефіцієнти кожній слабкій моделі на основі їх продуктивності, причому кращі моделі отримують вищі ваги. Це зважування дозволяє алгоритму визначати пріоритетність моделей, які більше сприяють зменшенню загальної помилки.

Щоб робити прогнози, ADABoostRegressor поєднує прогнози всіх слабких моделей, зважуючи їх відповідно до їх продуктивності під час навчання. Остаточний прогноз виходить шляхом агрегування зважених прогнозів усіх слабких моделей.

ADABoostRegressor має кілька переваг. Він ефективний у вирішенні складних задач регресії та може фіксувати нелінійні зв’язки між функціями та цільовою змінною. Він також менш схильний до переобладнання. Крім того, ADABoostRegressor відносно простий у реалізації та налаштуванні, і він надає оцінки важливості функцій, які можуть допомогти зрозуміти відносну важливість функцій.

Однак ADABoostRegressor може бути чутливим до шумних даних і викидів. Це також може вимагати ретельного налаштування гіперпараметрів для досягнення

оптимальної продуктивності. Крім того, процес навчання може бути обчислювально дорогим, особливо з великою кількістю ітерацій або складних слабких моделей.

Загалом ADABoostRegressor — це потужний алгоритм для завдань регресії, особливо в поєднанні зі слабкими моделями, які спеціалізуються на захопленні конкретних шаблонів у даних. Він зазвичай використовується в таких сферах, як фінанси, маркетинг і охорона здоров'я, де важливі точні прогнози.

GradientBoostingRegressor

GradientBoostingRegressor – це алгоритм машинного навчання, який належить до сімейства ансамблевого навчання. Він в основному використовується для завдань регресії, де метою є прогнозування безперервного числового значення на основі набору вхідних ознак.

Алгоритм працює шляхом поєднання кількох слабких моделей регресії, часто дерев рішень, у модель ансамблю. Він ітеративно будує ці моделі в послідовний спосіб, де кожна наступна модель зосереджується на зменшенні помилок, зроблених попередніми моделями. Цей процес відомий як посилення градієнта.

SVR

SVR означає опорну векторну регресію. Це алгоритм машинного навчання, який використовується для регресійних завдань, метою яких є прогнозування постійної цільової змінної.

У SVR мета полягає в тому, щоб знайти гіперплощину, яка найкраще відповідає навчальним даним, максимізуючи запас, подібно до SVM. Однак у SVR основна увага зосереджена на пошуку гіперплощини, яка допускає певний допуск або межу похибки. Цей допуск контролюється параметром, який називається епсилон.

SVR працює шляхом відображення вхідних функцій у більш вимірний простір функцій за допомогою функції ядра. У цьому трансформованому просторі SVR знаходить гіперплощину, яка найкраще розділяє точки даних у межах допуску. Опорні

вектори, які є точками даних, найближчими до запасу, відіграють вирішальну роль у визначенні регресійної моделі.

Модель SVR спрямована на мінімізацію емпіричного ризику, який є сумою помилок між прогнозованими та фактичними значеннями, а також враховує термін регуляризації для контролю складності моделі та запобігання переобладнанню.

SVR особливо корисний, коли маємо справу з проблемами нелінійної регресії або коли дані мають складні взаємозв'язки. Він може працювати з великими просторами функцій і стійкий до викидів.

XGBoost

XGBoost - це алгоритм машинного навчання, який використовується для задач класифікації, регресії та ранжування. Він базується на алгоритмі градієнтного бустингу і є одним з найбільш популярних алгоритмів у світі даних.

XGBoost має кілька особливостей, які дозволяють йому досягати високої точності прогнозування:

1) Розрахунок ваг для кожної змінної в залежності від її важливості.
2) Використання стохастичного градієнтного спуску, що дозволяє розв'язувати проблему перенавчання.
3) Підтримка розподілених обчислень, що дозволяє навчати модель на великих наборах даних.

MLPRegressor

MLRegressor розшифровується як Multi-Layer Perceptron Regressor. Це тип моделі нейронної мережі, який зазвичай використовується для завдань регресії.

MLPRegressor – це потужна та гнучка модель для задач регресії, здатна вивчати складні зв'язки в даних. Однак для забезпечення оптимальної продуктивності моделі важливо правильно налаштувати її гіперпараметри, виконати попередню обробку даних і врахувати можливість переобладнання.

1.4 Налаштування гіперпараметрів

Гіперпараметри визначають конфігурацію моделі і впливають на її продуктивність. Деякі загальні кроки для налаштування гіперпараметрів включають:

1) Вибір діапазону. Визначити діапазон можливих значень для кожного гіперпараметра. Наприклад, для швидкості навчання можна вибрати діапазон від 0.001 до 0.1.
2) Метод оптимізації. Вибрити метод оптимізації для пошуку найкращих значень гіперпараметрів. Це може бути перебір по сітці (grid search), випадковий пошук (random search) або більш складні методи – оптимізація з використанням алгоритмів на основі градієнта (gradient-based optimization) або алгоритмів на основі еволюції (evolutionary algorithms).
3) Крос-валідація. Використати для оцінки продуктивності моделі з різними значеннями гіперпараметрів. Крос-валідація дозволяє оцінити модель на різних підмножинах даних та забезпечує більш об'єктивну оцінку її продуктивності.
4) Метрики оцінки. Визначити метрики оцінки. Наприклад, для задачі регресії можуть використовуватись середня абсолютна помилка (MAE), середня квадратична помилка (MSE) або коефіцієнт детермінації (R2 score).
5) Пошук найкращих значень. Виконати пошук по заданому діапазону значень гіперпараметрів з використанням обраного методу оптимізації та оцінки продуктивності за допомогою крос-валідації. Знайти набір гіперпараметрів, який дає найкращі результати за визначеними метриками оцінки.
6) Аналіз результатів. Оцінити результати та зробити висновки щодо найкращих значень гіперпараметрів. Розглянути вплив різних значень гіперпараметрів на продуктивність моделі та визначити оптимальну конфігурацію для задачі.

1.5 Оцінка моделі

Оцінка моделі є важливим етапом після навчання моделі і включає аналіз її продуктивності та точності. Це допомагає зрозуміти, наскільки добре модель працює і наскільки точні її прогнози. Основні методи оцінки моделі включають:

- **R2-коефіцієнт** визначає частку варіації залежної змінної, яку можна пояснити моделлю. Він приймає значення від 0 до 1, де 1 означає ідеальне приближення до даних, а значення менше 0 означає, що модель виконує гірше, ніж просте середнє значення.
- **Adjusted R2 або Adjusted R-Squared** — це модифікована версія R2, яка враховує кількість незалежних змінних і розмір вибірки. Дозволяє порівняти продуктивність моделей з різною кількістю змінних та вибрати оптимальну модель для задачі регресії. Скориговане значення R2 також знаходиться в межах від 0 до 1, і більше значення вказує на кращу відповідність моделі. Adjusted R-Squared може бути обчислений за наступною формулою:

  Adjusted R-Squared = 1 - (1 - R-Squared) * (n - 1) / (n - k - 1),

  де R-Squared - коефіцієнт детермінації,

  n - кількість спостережень,

  k - кількість незалежних змінних.

  Adjusted R-Squared дозволяє порівняти продуктивність моделей з різною кількістю змінних та вибрати оптимальну модель для задачі регресії.
- **Середньоквадратична помилка (Mean Squared Error, MSE)** вимірює середнє квадратичне відхилення між прогнозованими значеннями та фактичними значеннями. Чим нижче значення MSE, тим краще модель.
- **RMSE (Root Mean Squared Error)** є квадратним коренем із MSE. Розраховується так, щоб він мав той самий масштаб, що й цільова змінна, і забезпечує більш сприйнятливу міру середньої помилки передбачення.

– **Середня абсолютна помилка (Mean Absolute Error, MAE)** вимірює середню абсолютну відхиленість між прогнозованими значеннями та фактичними значеннями. Чим нижче значення MAE, тим краще модель.

1.6 Опис предметної області

Для прогнозування використовуються різні методи.

Традиційні статистичні методи: методи, такі як регресійний аналіз та аналіз часових рядів, використовуються для прогнозування цін на ноутбуки на основі історичних даних.

Методи машинного навчання:

– регресійні моделі, такі як лінійна регресія та градієнтний бустінг, дозволяють побудувати складніші моделі для прогнозування цін на ноутбуки з високою точністю;
– існують також інші методи машинного навчання, такі як класифікація, кластеризація тощо, які можуть бути корисними для аналізу даних про ноутбуки та їх класифікації за ціновими сегментами або типами моделей.

Інтеграція з бізнес-аналітикою: важливо також враховувати потреби бізнесу та використовувати інструменти бізнес-аналітики для забезпечення зручного візуального аналізу результатів прогнозування цін на ноутбуки та виведення рекомендацій для прийняття рішень.

Переваги та обмеження існуючих підходів

Переваги:

– машинне навчання дозволяє автоматизувати процес аналізу та прогнозування цін на ноутбуки з високою точністю;
– інтеграція різних інструментів та методів дозволяє отримувати більш комплексні та точні результати;

– застосування новітніх технологій, таких як нейронні мережі, може покращити ефективність прогнозування.

Обмеження:

– Для успішного застосування методів машинного навчання потрібні великі обсяги даних та достатня обчислювальна потужність.
– Важливо правильно підібрати дані для аналізу та враховувати потенційні спотворення результатів від шуму або відсутності важливих факторів.
– Моделі машинного навчання можуть бути складними для розуміння та інтерпретації, що може ускладнювати прийняття рішень на практиці.

Загалом, існуючі інформаційні технології та методи машинного навчання вже дозволяють ефективно аналізувати та прогнозувати ціни на ноутбуки, але постійний розвиток та вдосконалення цих методів є важливим для досягнення ще більшої точності та ефективності.

1.7 Специфікація вимог до програмного забезпечення

1 ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

1.1 Призначення системи

Розробка програмного забезпечення для прогнозування цін на ноутбуки з використанням методів машинного навчання.

1.2 Погодження, що ухвалені в програмній документації

Документація повинна бути узгоджена з усіма учасниками проєкту перед початком розробки.

1.3 Межі проєкту ПЗ

Розробка обмежена сферою прогнозування цін на ноутбуки та відсутністю розширення на інші товари.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Сфера застосування

Програмне забезпечення призначене для аналізу та прогнозування цін на ноутбуки для допомоги користувачам у прийнятті інформованих рішень.

2.2 Характеристики користувачів

Кінцеві користувачі - фахівці з аналізу ринку електроніки.

2.3 Загальна структура і склад системи

Система складається з інтерфейсу користувача, модуля обробки даних та модуля машинного навчання.

2.4 Загальні обмеження

Програмне забезпечення обмежене доступом до даних з інтернету та обробкою даних про ноутбуки.

3 ФУНКЦІЇ СИСТЕМИ

3.1 Збір даних

3.1.1 Опис функції

Забезпечення можливості завантаження даних з різних джерел, таких як інтернет-ресурси та бази даних.

3.2.2 Вхідна і вихідна інформація

Вхідна інформація: файл laptops.csv.

Вихідна інформація: завантажені дані для подальшого аналізу.

3.2.3 Функціональні вимоги

Система повинна підтримувати такі формати даних для завантаження: .csv, .xlsx, .db, .sqlite.

Система повинна відображати статус завантаження даних.

3.2 Описовий аналіз

3.2.1 Опис функції

Виведення основних статистичних показників та візуалізація даних.

3.2.2 Вхідна і вихідна інформація

Вхідна інформація: завантажені дані для аналізу.

Вихідна інформація: графіки, діаграми, статистичні показники.

3.2.3 Функціональні вимоги

Система повинна надавати можливість вибору параметрів для аналізу.

Система повинна підтримувати візуалізацію різних типів даних.

4 ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Джерела і зміст вхідної інформації (даних)

4.1.1 Вхідна інформація

Параметри ноутбуків, ціни, характеристики.

4.1.2 Вихідна інформація

Очищені та нормалізовані дані для аналізу.

4.2 Нормативно-довідкова інформація

4.2.1 Класифікатори

Використання класифікаторів для категоризації товарів та інших стандартів.

4.2.2 Довідники

Застосування довідників для однозначного визначення характеристик.

4.3 Вимоги до способів організації, збереження та ведення інформації

4.3.1 Організація

Дані повинні зберігатися у форматі бази даних.

4.3.2 Збереження

Застосування механізмів резервного копіювання та збереження даних.

5 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Процесор: Intel Core i5 або еквівалентний.

Оперативна пам'ять: не менше 8 ГБ.

Вільне місце на жорсткому диску: не менше 20 ГБ.

6 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

6.1 Архітектура програмної системи

6.1.1 Опис архітектури

Система буде побудована на основі клієнт-серверної архітектури, де клієнтська частина буде взаємодіяти з користувачем, а серверна частина відповідатиме за обробку та аналіз даних.

6.1.2 Компоненти системи

Компоненти системи включатимуть інтерфейс користувача, модуль обробки даних та модель машинного навчання.

6.2 Системне програмне забезпечення

6.2.1 Операційна система

Розробка буде проводитися для операційних систем Windows.

6.3 Мережне програмне забезпечення

6.3.1 Комунікаційні протоколи

Взаємодія між клієнтом та сервером буде здійснюватися за допомогою протоколів HTTP та REST.

6.3.2 Безпека мережі

Забезпечення захищеного каналу зв'язку шляхом використання шифрування засобами протоколу HTTPS.

6.4 Програмне забезпечення ведення інформаційної бази

6.4.1 Система управління базами даних (СУБД)

Використання СУБД SQL для зберігання та управління даними.

6.4.2 Засоби резервного копіювання

Забезпечення автоматичного резервного копіювання бази даних щоденно з можливістю відновлення.

6.5 Мова і технологія розробки ПЗ

6.5.1 Вибір мови програмування

Використання мови програмування Python для розробки клієнтської та серверної частин програмного забезпечення.

6.5.2 Технології розробки

Використання фреймворків Django та Flask для розробки серверної частини. Використання бібліотек машинного навчання, таких як Scikit-learn, для аналізу даних та прогнозування.

Висновки до розділу 1

У результаті написання розділу 1 КРМ було виконано завдання для досягнення мети, яка полягала в покращенні точності прогнозування цін ноутбуків за рахунок розробки застосунку для аналізу великого набору даних із використанням методів машинного навчання.

У розділі 1 була надана теоретична основа щодо машинного навчання та поставлена задача для досягнення поставленої мети та було визначено кроки навчання моделі, до яких входить:

- Розбиття даних.
- Вибір моделі.
- Навчання моделі.
- Налаштування гіперпараметрів.
- Оцінка моделі.

Також було описано різні регресори та гіперпараметри, які потрібні для їх реалізації. Описано предметну область та специфікацію вимог.

2 ПЕРЕДОБРОБКА ТА ПЕРВИННИЙ АНАЛІЗ ДАНИХ

2.1 Опис набору даних та ознак

Це набір даних [2] про брухт ноутбуків від Flipkart, який містить інформацію про різні аспекти ноутбуків, такі, як ціни, знижки, технічні характеристики та гарантію. Набір даних містить загалом 920 записів, кожна з яких представляє один ноутбук. Дані впорядковано в 10 стовпців.

Цільовою змінною є ціна на ноутбук на основі його характеристик, категоріальна змінна, яка включає в себе характеристики ноутбуку.

Всі характеристики та їх опис наведені у таблиці 2.1.

Таблиця 2.1 – Опис характеристик ноутбуків

Ознака (англійською)	**Ознака (українською)**	**Пояснення**
title	назва	Короткий опис
price	ціна	Вартість
discount	знижка	Будь-які діючі знижки на ціну
Processor	Процесор	Тип процесора
RAM	Оперативна пам’ять	Обсяг оперативної пам’яті
OS	Операційна система	Операційна система
SSD	Твердотільний накопичувач	Розмір твердотільного накопичувача
Display	Дисплей	Розмір екрана та технічні характеристики дисплея
In_build_sw	Вбудоване програмне забезпечення	Перераховано будь-яке програмне забезпечення, попередньо встановлене
warranty	гарантія	Інформація про гарантію, що надається

Важливо зауважити, що в наборі даних відсутні деякі значення для деяких стовпців, зокрема «discount», «In_build_sw» і «warranty». Це слід враховувати при

аналізі даних. Крім того, усі стовпці мають тип даних «об’єкт», який може потребувати подальшої обробки для перетворення даних у придатний для використання формат.

2.2 Первинний аналіз даних

Первинний аналіз даних - це процес огляду та розуміння набору даних перед подальшою обробкою та аналізом. Він допомагає виявити загальну структуру даних, їх характеристики, аномалії та потенційні проблеми, що потребують уваги.

Основними кроками для проведення первинного аналізу даних є: огляд даних, статистичний аналіз, обробка відсутніх даних, виявлення потенційних викидів або аномалій та перевірка унікальних значень.

Огляд даних. Переглянути загальну структуру набору даних, включаючи кількість рядків та стовпців, назви стовпців та їх типи (числові, категоріальні тощо).

Статистичний аналіз. Обчислити основні статистичні показники для числових стовпців, такі як середнє значення, медіану, мінімум, максимум, стандартне відхилення. Це дозволить отримати загальну уяву про розподіл даних та їх числові характеристики.

Обробка відсутніх даних. Виявити наявність відсутніх даних та розглянути можливі варіанти їх обробки. Це може включати видалення рядків або стовпців з відсутніми даними, заповнення відсутніх значень середніми або медіанними значеннями, або використання інших методів, що підходять до конкретного випадку.

Виявлення потенційних викидів або аномалій. Перевірити дані на наявність потенційних викидів або аномальних значень. Це можна зробити, використовуючи статистичні методи, такі як виявлення викидів на основі стандартного відхилення або використовуючи графіки розсіювання. Виявлення таких аномалій дозволяє виявити можливі помилки в даних або особливості, які потребують подальшого дослідження.

Перевірка унікальних значень. Перевірити унікальні значення для кожної змінної, щоб переконатися, що дані відповідають очікуваному формату та діапазону. Це допоможе виявити потенційні помилки або некоректні дані.

Ці кроки первинного аналізу даних допоможуть зрозуміти загальну структуру, особливості та потенційні проблеми в наборі даних. А також ідентифікувати основні характеристики та потенційні проблеми. Він служить важливою підготовчою стадією перед подальшим аналізом даних, дозволяючи виявити ключові аспекти, які можуть бути використані для подальших висновків або моделювання.

Огляд даних. Перед оглядом даних спочатку потрібно встановити всі потрібні бібліотеки та підключити файл із базою даних.

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("./sample_data/laptops.csv",  index_col=0)
```

У коді представлений імпорт бібліотек, які потрібні на даному етапі роботи.

Далі представлено огляд даних (рисунок 2.1).

```
df.head()
```

	title	price	discount	Processor	RAM	OS	SSD	Display	In_build_sw	warranty
0	DELL Vostro Core i3 11th Gen - (8 GB/1 TB HDD/...	₹37,990	₹58,48935% off	Processor: Intel i3-1115G4 (Base- 1.7 GHz & Tu...	RAM & Storage: 8GB DDR4 & 1TB HDD + 256GB SSD	Graphics & Keyboard: Integrated & Standard Key...	Display: 15.6" FHD WVA AG Narrow Border	Intel Core i3 Processor (11th Gen)	NaN	8 GB DDR4 RAM
1	HP 14s Intel Core i3 11th Gen - (8 GB/256 GB S...	₹35,490	₹47,20624% off	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	256 GB SSD	35.56 cm (14 inch) Display	NaN	1 Year Onsite Warranty
2	Lenovo V15 G2 Core i3 11th Gen - (8 GB/512 GB ...	₹33,999	₹59,76043% off	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	No	1 Year Onsite Warranty + 1 Year Accidental Dam...
3	HP 15s Intel Core i3 12th Gen - (8 GB/512 GB S...	₹45,490	₹56,26019% off	Intel Core i3 Processor (12th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	Microsoft Office Home & Student 2021	1 Year Onsite Warranty
4	ASUS VivoBook 15 (2022) Core i3 10th Gen - (8 ...	₹33,990	₹45,99026% off	Intel Core i3 Processor (10th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 inch) Display	Office Home and Student 2021	1 Year Onsite Warranty

Рисунок 2.1 – Огляд даних

Деякі ознаки будуть розбиті на декілька для більш коректного відображення.

Також представлено кількість рядків та стовпців.

```
df.shape
```

Результат: (920, 10).

Таким чином, датафрейм містить 920 записів та 10 ознаки. Поглянемо детальну інформацію про ознаки, а саме назви стовпців та їх типи.

```
df.info()
```

Результат представлено на рисунку 2.2.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 920 entries, 0 to 919
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   title        920 non-null    object
 1   price        920 non-null    object
 2   discount     831 non-null    object
 3   Processor    920 non-null    object
 4   RAM          920 non-null    object
 5   OS           920 non-null    object
 6   SSD          920 non-null    object
 7   Display      920 non-null    object
 8   In_build_sw  357 non-null    object
 9   warranty     906 non-null    object
dtypes: object(10)
memory usage: 79.1+ KB
```

Рисунок 2.2 – Інформація про ознаки

Під час перегляду інформації про ознаки було виявлено пропуски, тому далі вони будуть оброблені.

Статистичний аналіз. Для отримання загальної уяви про розподіл даних та їх числові характеристики потрібно обчислити основні статистичні показники для числових стовпців, такі як середнє значення, медіану, мінімум, максимум, стандартне відхилення. Даний вигляд бази даних не дозволяє цього зробити, оскільки всі колонки є категоріальними.

Тому було переглянуто статистику для нечислових ознак (рисунок 2.3).

```
df.describe()
```

	title	price	discount	Processor	RAM	OS	SSD	Display	In_build_sw	warranty
count	920	920	831	920	920	920	920	920	357	906
unique	846	469	763	92	57	45	52	88	118	117
top	Infinix X1 Slim Series Core i5 10th Gen - (8 G...	₹54,990	₹74,99026% off	Intel Core i5 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 inch) Display	Office Home and Student 2021	1 Year Onsite Warranty
freq	4	17	4	95	357	436	496	253	42	456

Рисунок 2.3 – Нечислові ознаки

Перевірка унікальних значень. Перевірка унікальних значень допомагає виявити потенційні помилки або некоректні дані. На наступних рисунках перевіряється унікальність різних характеристик ноутбуків.

```
# Checking the values in the column 'warranty'
df.Warranty.unique()
```

```
array(['8 GB DDR4 RAM', '1 Year Onsite Warranty',
       '1 Year Onsite Warranty + 1 Year Accidental Damage Protection Add On',
       '1 Year onsite warranty', '2 Years Onsite\ufffdWarranty',
       '1 Year International Travelers Warranty (ITW)',
       '1 Year Limited Warra\xadnty',
       'Intel Core i3 Processor (11th Gen)', '16 GB DDR4 RAM',
       '1 Year Onsite Warranty + 1 Year Premium Care + 1 Year Accidentaly Damage Protection',
       '2 Year Onsite\ufffdWarranty',
       '1 Year Onsite\ufffdWarranty\ufffd+ 1 Year Premium Care + 1 Year Accidental Damage Protection',
       '1 Year Warranty', '2 Year Carry-In Warranty Term',
       '1 Year Carry-in Warranty',
       '1 Year International Travelers Warranty',
       '1 Year Limited Warranty', '1 Year Onsite Hardware Service',
       '1 Year Manufacturer', '2 Year Onsite Warranty', nan,
       '2 Year On-Site & Carry-In Warranty',
       '1 Year On-Site & Carry-In Warranty', '1 Year Onsite warranty',
       '720p HD webcam with physical privacy shutter',
       'WiFi & BT: 802.11ac 1x1 WiFi and Bluetooth',
       '1 Year Onsite\ufffdWarranty\ufffd+ 1 Year Legion Ultimate Support + 1 Year Accidental Damage Protection',
       '1 YEAR', '64 bit Windows 11 Operating System',
       'AMD Ryzen 7 Octa Core Processor', '39.62 cm (15.6 inch) Display',
       '39.62 cm (15.6 inches) Display',
       '64 bit Windows 10 Operating System',
       '1 Year Onsite Warranty + 1 Year Premium Care + 1 Year Accidental Damage Protection',
       '16 GB DDR5 RAM', '2 Years Onsite Warranty',
       '1 Year onsite Warranty', '12 Months', '1 Year Premium Support',
       '12 months',
       'Ports: 1. HiSpeed USB 2.0 | 2. HiSpeed USB 2.0 with PowerShare | 3. USB 3.2 Gen 1 Type-C port with
DisplayPort with alt mode | 4. SuperSpeed USB 3.2 | 5. HDMI | 6. Power in | 7. RJ45 | 8. Headphones/mic',
       '1 Year On-Site Warranty', '1 YEAR WAARANTY', '1 Year',
       '1 Years Carry in Warranty', 'Intel Core i7 Processor (10th Gen)',
       'AMD Ryzen 5 Hexa Core Processor',
       '1 Year Warranty + 1 Year Premium Care + 1 Year ADP',
       '3 Years Onsite\ufffdWarranty\ufffd+ 3 Years Legion Ultimate Support + 1 Year Accidental Damage
Protection',
       '2 Year Carry-in Warranty',
       'Ports: (1) HDMI 2.1, (2) SuperSpeed USB 2.0 Gen 1 Type-A including (1) with PowerShare, (1)
SuperSpeed USB 3.2, (1) USB-C Data/Display Port Alt-Mode, Headphone/Mic, (1) RJ45',
       '1 Years Warranty', '1 year Onsite Warranty',

✓ 13 сек.  выполнено в 17:38
```

Рисунок 2.4 – Перевірка унікальності «Гарантій»

```
# Checking the different processors
df.Processor.unique()
```

```
array(['Processor: Intel i3-1115G4 (Base- 1.7 GHz & Turbo up to 4.10 GHz) 2 Cores',
       'Intel Core i3 Processor (11th Gen)',
       'Intel Core i3 Processor (12th Gen)',
       'Intel Core i3 Processor (10th Gen)',
       'AMD Ryzen 7 Octa Core Processor',
       'AMD Ryzen 5 Hexa Core Processor',
       'Intel Core i5 Processor (12th Gen)',
       'AMD Ryzen 3 Dual Core Processor',
       'Intel Core i5 Processor (11th Gen)',
       'Intel Celeron Dual Core Processor',
       'Intel Celeron Quad Core Processor',
       'AMD Athlon Dual Core Processor',
       'AMD Ryzen 5 Quad Core Processor', 'Apple M1 Processor',
       'Processor: Intel i3-1115G4 (Base- 1.70 GHz & Turbo up to 4.10 GHz) 2 Cores',
       'Processor: Intel i5-1235U (Base- 3.30 GHz & Turbo up to 4.40 GHz) 10 Cores',
       'Intel Core i7 Processor (12th Gen)',
       'Intel Core i7 Processor (11th Gen)',
       'AMD Ryzen 3 Quad Core Processor',
       'Qualcomm Snapdragon 7c Gen 2 Processor', 'Apple M2 Processor',
       'Apple M1 Max Processor',
       'Processor-i3-1115G4 Processor upto 4.1 GHz Speed',
       'Intel Core i5 Processor (10th Gen)', 'AMD Dual Core Processor',
       'Get unparalleled power and reliability with the new Intel 12th Gen CPU',
       'Processor: AMD Ryzen 3-3250U (2.60 GHz up to 3.50 GHz)',
       'Apple M2 Pro Processor', 'NVIDIA RTX 3050 Graphics upto 90W TGP',
       'Processor: R3-5425U (2.70 GHz up to 4.1 GHz)',
       'Processor: AMD Ryzen 7-5825U (2.00 GHz up to 4.50 GHz)',
       'Intel Pentium Silver Processor',
       'Free upgrade to Windows 11 when available',
       'Stylish & Portable Thin and Light Laptop',
       'Pre-installed Genuine Windows 10 Home OS',
       'Powered by 11th Gen Intel Evo Core i5 Processor',
       'AMD Ryzen 9 Octa Core Processor',
       'AMD Ryzen 7 Hexa Core Processor',
       'Processor: Intel i7-1255U (Base- 3.50 GHz & Turbo up to 4.70 GHz) 10 Cores',
       'Intel Core i9 Processor (12th Gen)',
       'AMD Ryzen 3 Dual Core Processor (3rd Gen)',
       'Apple M1 Pro Processor',
       'Processor: Intel i7-11800H- (2.30 GHz up to 4.60 GHz) 16MB L3',
       'Processor: Intel i9-12900H (Base- 3.80 GHz & Turbo up to 5.0 GHz) 14 Cores',
       'AMD Ryzen 9 Octa Core Processor (5th Gen)',
       'Intel Core i7 Processor (10th Gen)', 'NVIDIA GeForce RTX 2060',
```

✓ 13 сек. выполнено в 17:38

Рисунок 2.5 – Перевірка унікальності «Процесорів»

```
df.SSD_Size.unique()
```

Результат: `array([ nan, 0.25 , 0.5 , 0.125, 1. , 2. , 4. ])`.

```
df.Display.unique()
```

Результат: `array(['Not specified', '14', '15.6', '13.3', '17.3', '16', '16.2', '15', '14.1', '16.1', '14.2', '13.4', '14.5', '13.5', '13', '12'], dtype=object).`

2.3 Передобробка даних

Передобробка даних є важливим етапом в аналізі даних, оскільки дозволяє підготувати дані для подальшого застосування алгоритмів машинного навчання. Вона включає в себе ряд кроків, які допомагають зробити дані зрозумілішими, консистентними та готовими до використання. Основні кроки передобробки даних включають:

1) Видалення дублікатів. Перевірка наявності та видалення повторюваних записів в наборі даних.
2) Обробка пропущених значень. Аналіз та обробка відсутніх значень в даних. Це може включати заповнення пропущених значень середніми, медіанними або найближчими значеннями, або видалення відповідних записів.
3) Видалення непотрібних змінних. Видалення зайвих змінних, які не мають впливу на аналіз або прогнозування.
4) Кодування категоріальних змінних. Перетворення категоріальних змінних у числовий формат, щоб їх можна було використовувати в алгоритмах машинного навчання.
5) Масштабування змінних. Нормалізація або стандартизація числових змінних для забезпечення однакового масштабу та уникнення впливу величини значень на алгоритми машинного навчання.
6) Видалення викидів. Виявлення та видалення викидів або аномальних значень, які можуть спотворити аналіз.
7) Вибір ознак. Відбір найбільш важливих ознак або змінних для покращення ефективності моделі.

Ці кроки передобробки даних допомагають забезпечити якість та достовірність даних перед їх використанням.

Важливо пам'ятати, що передобробка даних є ітеративним процесом, і після застосування методів може знадобитися перевірка та корекція результатів. Крім того, варто зберігати оригінальні дані, щоб мати можливість повернутися до них при необхідності.

Правильна передобробка даних гарантує, що модель глибинного машинного навчання буде працювати на якісних та репрезентативних даних, що забезпечує точність та надійність аналізу та прогнозування.

Перед обробкою потрібно перейменувати деякі стовпці для узгодженості та більш коректного відображення.

```
df.rename(columns = {'title':'Title'}, inplace = True)
df.rename(columns = {'price':'Price'}, inplace = True)
df.rename(columns = {'In_build_sw':'InBuild_Software'}, inplace = True)
df.rename(columns = {'warranty':'Warranty'}, inplace = True)
```

Потрібно видалити стовпчик, який не буде використовуватись.

```
df = df.drop(['discount'], axis=1)
```

Для обробки відсутніх значень використовується заповнення даними.

```
# Categorising the column 'In_build_sw' in two ways,
#whether the laptop has any software or not.
df.InBuild_Software = df.InBuild_Software.fillna('No')
df.InBuild_Software = np.where(df.InBuild_Software == 'No', 'No', 'Yes')
```

Результат представлено на рисунку 2.6.

	Title	Price	Processor	RAM	OS	SSD	Display	InBuild_Software	Warranty
0	DELL Vostro Core i3 11th Gen - (8 GB/1 TB HDD/...	₹37,990	Processor: Intel i3-1115G4 (Base- 1.7 GHz & Tu...	RAM & Storage: 8GB DDR4 & 1TB HDD + 256GB SSD	Graphics & Keyboard: Integrated & Standard Key...	Display: 15.6" FHD WVA AG Narrow Border	Intel Core i3 Processor (11th Gen)	No	No
1	HP 14s Intel Core i3 11th Gen - (8 GB/256 GB S...	₹35,490	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	256 GB SSD	35.56 cm (14 inch) Display	No	1.0
2	Lenovo V15 G2 Core i3 11th Gen - (8 GB/512 GB ...	₹33,999	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	No	1.0
3	HP 15s Intel Core i3 12th Gen - (8 GB/512 GB S...	₹45,490	Intel Core i3 Processor (12th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	Yes	1.0
4	ASUS VivoBook 15 (2022) Core i3 10th Gen - (8 ...	₹33,990	Intel Core i3 Processor (10th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 inch) Display	Yes	1.0
...	...	...	...	...	...	...	...	...	...
915	Lenovo Intel Core i7 12th Gen - (16 GB/512 GB ...	₹1,19,990	Intel Core i7 Processor (12th Gen)	16 GB LPDDR5 RAM	Windows 11 Operating System	512 GB SSD	35.56 cm (14 Inch) Touchscreen Display	No	3.0

Рисунок 2.6 – Результат

Нормалізація колонки «Гарантія» шляхом прибирання зайвої інформації.

```
# Getting the years of warranty
# Removed the values which did not have any year specified in it
# filtered out the years for the rest of values.
df.Warranty = np.where(df.Warranty.str.contains('Year') == False,
                       'No',df.Warranty.str.split(' ').str[0])
```

```
df.Warranty = np.where(~df.Warranty.isin(['1','2','3']), 'No',
                       df.Warranty.str.split(' ').str[0])
df.Warranty = pd.to_numeric(df.Warranty, errors='coerce')
# Filled the NaN values
df.Warranty = df.Warranty.fillna('No')
df.Warranty.unique()
df
```

Результат представлено на рисунку 2.7.

	Title	Price	Processor	RAM	OS	SSD	Display	InBuild_Software	Warranty
0	DELL Vostro Core i3 11th Gen - (8 GB/1 TB HDD/...	₹37,990	Processor: Intel i3-1115G4 (Base- 1.7 GHz & Tu...	RAM & Storage: 8GB DDR4 & 1TB HDD + 256GB SSD	Graphics & Keyboard: Integrated & Standard Key...	Display: 15.6" FHD WVA AG Narrow Border	Intel Core i3 Processor (11th Gen)	No	No
1	HP 14s Intel Core i3 11th Gen - (8 GB/256 GB S...	₹35,490	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	256 GB SSD	35.56 cm (14 inch) Display	No	1.0
2	Lenovo V15 G2 Core i3 11th Gen - (8 GB/512 GB ...	₹33,999	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	No	1.0
3	HP 15s Intel Core i3 12th Gen - (8 GB/512 GB S...	₹45,490	Intel Core i3 Processor (12th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	Yes	1.0
4	ASUS VivoBook 15 (2022) Core i3 10th Gen - (8 ...	₹33,990	Intel Core i3 Processor (10th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 inch) Display	Yes	1.0
...	...	...	...	...	...	...	...	...	...
915	Lenovo Intel Core i7 12th Gen - (16 GB/512 GB ...	₹1,19,990	Intel Core i7 Processor (12th Gen)	16 GB LPDDR5 RAM	Windows 11 Operating System	512 GB SSD	35.56 cm (14 Inch) Touchscreen Display	No	3.0

Рисунок 2.7 – Нормалізація колонки «Гарантія»

Нормалізація колонки «Ціна» представлено на рисунку 2.8.

```
# Transforming the price
# Removing the comma in between and removing the currency symbol
df.Price = df.Price.str.replace('₹', '')
df.Price = df.Price.str.replace(',','')
df.Price = pd.to_numeric(df.Price, errors='coerce')
df.Price
```

```
0       37990
1       35490
2       33999
3       45490
4       33990
        ...
915    119990
916     68990
917     35990
918     36990
919     89999
Name: Price, Length: 920, dtype: int64
```

Рисунок 2.8 – Нормалізація колонки «Ціна»

Виділення брендів ноутбуків у окрему колонку із колонки «Назва» представлено на рисунку 2.9).

```
# Extracting the brands of the laptops from the title
df['Laptop_Brands'] = df.Title.str.split(' ').str[0]
df['Laptop_Brands'].value_counts()
```

Title	Price	Processor	RAM	OS	SSD	Display	InBuild_Software	Warranty	Laptop_Brands
DELL) Core h Gen GB/1 IDD/...	37990	Intel i3-1115G4 (Base- 1.7 GHz & Turbo up to 4...	RAM & Storage: 8GB DDR4 & 1TB HDD + 256GB SSD	Graphics & Keyboard: Integrated & Standard Key...	Display: 15.6" FHD WVA AG Narrow Border	Intel Core i3 Processor (11th Gen)	No	No	DELL
s Intel 3 11th en - (8 56 GB S...	35490	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	256 GB SSD	35.56 cm (14 inch) Display	No	1.0	HP
o V15 ore i3 Gen - B/512 GB ...	33999	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	No	1.0	Lenovo
s Intel ore i3 Gen - B/512 B S...	45490	Intel Core i3 Processor (12th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	Yes	1.0	HP
ASUS oBook (2022) ore i3 Gen - (8 ...	33990	Intel Core i3 Processor (10th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 inch) Display	Yes	1.0	ASUS

Рисунок 2.9 – Колонка «Бренди»

Виділення окремої колонки «Бренд процесору» представлено на рисунку 2.10.

```
# Replacing 'Processor: ' in the beginning of some of the values
df.Processor = df.Processor.str.replace('Processor: ', '')
# Extracting the processor brands
df['Processor_Brands'] = df.Processor.str.split(' ').str[0]
df['Processor_Brands'].value_counts()
df
```

cessor	RAM	OS	SSD	Display	InBuild_Software	Warranty	Laptop_Brands	Processor_Brands
ntel i3-1115G4 se- 1.7 GHz & urbo up to 4...	RAM & Storage: 8GB DDR4 & 1TB HDD + 256GB SSD	Graphics & Keyboard: Integrated & Standard Key...	Display: 15.6" FHD WVA AG Narrow Border	Intel Core i3 Processor (11th Gen)	No	No	DELL	Intel
el Core i3 ocessor h Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	256 GB SSD	35.56 cm (14 inch) Display	No	1.0	HP	Intel
el Core i3 ocessor h Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	No	1.0	Lenovo	Intel
el Core i3 ocessor h Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 Inch) Display	Yes	1.0	HP	Intel
el Core i3 ocessor h Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD	39.62 cm (15.6 inch) Display	Yes	1.0	ASUS	Intel

Рисунок 2.10 – Колонка «Бренд процесору»

На рисунку 2.11 представлено виділення серії та покоління процесорів у окремі колонки з колонки «Процесор».

```
# Extracting the Processor Series and Generation from the Processors
# Intel Processor
df_copied = df.copy()
df = df.assign(Processor_Series = df_copied.Processor.str.split(' ')
               .str[2])
df.loc[(df.Processor_Brands == 'Intel')
       & ~(df.Processor_Series.isin(['i3','i5','i7','i9'])),
       ['Processor_Series']] = 'Not specified'
# Intel Generation
```

```
df = df.assign(Processor_Gen = df_copied.Processor.str.split('(')
               .str[1].str.replace(')','',regex=True))
df.loc[(df.Processor_Brands == 'Intel')
       & ~(df.Processor_Gen.isin(['3rd Gen', '4th Gen',
                                  '5th Gen', '7th Gen',
                                  '8th Gen', '9th Gen',
                                  '10th Gen', '11th Gen',
                                  '12th Gen'])),
       ['Processor_Gen']] = 'Not specified'
```

SSD	Display	InBuild_Software	Warranty	Laptop_Brands	Processor_Brands	Processor_Series	Processor_Gen
isplay: 15.6" FHD WVA AG Narrow Border	Intel Core i3 Processor (11th Gen)	No	No	DELL	Intel	Not specified	Not specified
56 GB SSD	35.56 cm (14 inch) Display	No	1.0	HP	Intel	i3	11th Gen
12 GB SSD	39.62 cm (15.6 Inch) Display	No	1.0	Lenovo	Intel	i3	11th Gen
12 GB SSD	39.62 cm (15.6 Inch) Display	Yes	1.0	HP	Intel	i3	12th Gen
12 GB SSD	39.62 cm (15.6 inch) Display	Yes	1.0	ASUS	Intel	i3	10th Gen

Рисунок 2.11 – Результат

Виділення розміру та типу пам’яті, а також розділення на SSD та HDD пам’ять представлено на рисунку 2.12. Також видалено зайві колонки.

```
# Extracting the RAM Size
df_copied = df.copy()
df = df.assign(RAM_Size = df_copied.RAM.str.split(' ').str[0])
df.loc[~(df.RAM_Size.isin(['4','8','16','32'])),
       ['RAM_Size']] = ''
df.RAM_Size = pd.to_numeric(df.RAM_Size, errors='coerce')
# Extracting the type of RAM
df = df.assign(RAM_Type = df_copied.RAM.str.split(' ').str[2])
df.loc[~(df.RAM_Type.isin(['DDR3', 'DDR4', 'DDR5',
                           'LPDDR3', 'LPDDR4', 'LPDDR4X',
                           'LPDDR5'])),['RAM_Type']] = 'Not specified'
# Creating a new column which contains the size of HDD in the laptops
df['HDD_Size'] = np.where(df.SSD.str.contains('HDD'),
                          df.SSD.str.split('HDD').str[0], 0)
hdd_options = {'1 TB ': 1,
               '256 GB ': 0.25,
               '512 GB ': 0.5,
               '2 TB ': 2}
df = df.replace({'HDD_Size': hdd_options})
# Removing '|' from the values
df.SSD = df.SSD.str.replace('|','', regex = True)
# Creating a new column which contains the size of SSD in the laptops
df['SSD_Size'] = np.where(df.SSD.str.contains('SSD'),
                          df.SSD.str.split('SSD').str[0], 0)
# Removing the HDD values from the SSD column
df.SSD_Size = df.SSD_Size.str.replace('.*HDD', '', regex = True)
ssd_options = {'1 TB ': 1,
               '256 GB ': 0.25,
               '512 GB ': 0.5,
               '2 TB ': 2,
               '4 TB ': 4,
               '128 GB ': 0.125}
df = df.replace({'SSD_Size': ssd_options})
```

Laptop_Brands	Processor_Brands	Processor_Series	Processor_Gen	RAM_Size	RAM_Type	HDD_Size	SSD_Size
DELL	Intel	Not specified	Not specified	8.0	DDR4	0.0	0.00
HP	Intel	i3	11th Gen	8.0	DDR4	0.0	0.25
Lenovo	Intel	i3	11th Gen	8.0	DDR4	0.0	0.50

Рисунок 2.12 – Нормалізація колонки «Пам’ять»

Нормалізація колонки «Дисплей» представлено на рисунку 2.13.

```
# Removing all the values which do not have any size in it.
df.loc[~df.Display.str.contains('inch'),'Display'] = 'Not specified'
# Extrcating the inches from the display size
df.Display = df.Display.str.split('(').str[1]
df.Display = df.Display.replace(' inch.*', '', regex = True)
#Filling NaN values with 'Not specified'
df.Display = df.Display.fillna(0)
df.Display.unique()
```

	Title	Price	OS	Display	InBuild_Software	Warranty	Laptop_Brands	Processor_Brands	Proce
0	DELL Vostro Core i3 11th Gen - (8 GB/1 TB HDD/...	37990	Other	0	No	No	DELL	Intel	
1	HP 14s Intel Core i3 11th Gen - (8 GB/256 GB S...	35490	Windows 11	14	No	1.0	HP	Intel	
2	Lenovo V15 G2 Core i3 11th Gen - (8 GB/512 GB ...	33999	Windows 11	0	No	1.0	Lenovo	Intel	

Рисунок 2.13 – Нормалізація колонки «Дисплей»

Нормалізація колонки «Операційна система» представлена на рисунку 2.14.

```
# Updating the OS with more suitable values.
options = {'64 bit Windows 11 Operating System' : 'Windows 11',
'64 bit Windows 10 Operating System' : 'Windows 10',
'Windows 11 Operating System' : 'Windows 11',
'Mac OS Operating System' : 'Mac OS',
'Windows 10 Operating System' : 'Windows 10',
'DOS Operating System' : 'DOS',
'Chrome Operating System' : 'Chrome',
'64 bit Chrome Operating System' : 'Chrome',
'32 bit Windows 11 Operating System' : 'Windows 11',
'64 bit Windows 8 Operating System' : 'Windows 8'}

df = df.replace({'OS': options})
df.loc[~(df.OS.isin(['Windows 8', 'Windows 10', 'Windows 11', 'Mac OS',
'DOS', 'Chrome'])),['OS']] = 'Other'
```

	Title	Price	OS	Display	InBuild_Software	Warranty	Laptop_Brands	Processor_Brands	Proce
0	DELL Vostro Core i3 11th Gen - (8 GB/1 TB HDD/...	37990	Other	Not specified	No	No	DELL	Intel	
1	HP 14s Intel Core i3 11th Gen - (8 GB/256 GB S...	35490	Windows 11	14	No	1.0	HP	Intel	
2	Lenovo V15 G2 Core i3 11th Gen - (8 GB/512 GB ...	33999	Windows 11	Not specified	No	1.0	Lenovo	Intel	

Рисунок 2.14 – Нормалізація колонки «Операційна система»

Після всіх перетворень можна перевірити числові ознаки (рисунок 2.15.

	RAM_Size	HDD_Size	SSD_Size	Price
count	873.00	836.00	807.00	889.00
mean	11.76	0.08	0.57	87960.21
std	5.81	0.27	0.32	69313.23
min	4.00	0.00	0.12	15990.00
25%	8.00	0.00	0.50	45990.00
50%	8.00	0.00	0.50	69092.00
75%	16.00	0.00	0.50	104990.00
max	32.00	2.00	4.00	1174131.00

Рисунок 2.15 – Числові ознаки

Перевірка даних перед початком візуального аналізу даних представлено на рисунку 2.16.

	Title	Price	OS	Display	InBuild_Software	Warranty	Laptop_Brands	Processor_Brands	Processor_Series	Processor_Gen	RAM_Size	RAM_Type	HDD_Size	SSD_Size
0	DELL Vostro Core i3 11th Gen - (8 GB/1 TB HDD/...	37990	Other	Not specified	No	No	DELL	Intel	Not specified	Not specified	8.0	DDR4	NaN	NaN
1	HP 14s Intel Core i3 11th Gen - (8 GB/256 GB S...	35490	Windows 11	14	No	1.0	HP	Intel	i3	11th Gen	8.0	DDR4	0.0	0.25
2	Lenovo V15 G2 Core i3 11th Gen - (8 GB/512 GB ...	33999	Windows 11	Not specified	No	1.0	Lenovo	Intel	i3	11th Gen	8.0	DDR4	0.0	0.50
3	HP 15s Intel Core i3 12th Gen - (8 GB/512 GB S...	45490	Windows 11	Not specified	Yes	1.0	HP	Intel	i3	12th Gen	8.0	DDR4	0.0	0.50
4	ASUS VivoBook 15 (2022) Core i3 10th Gen - (8 ...	33990	Windows 11	15.6	Yes	1.0	ASUS	Intel	i3	10th Gen	8.0	DDR4	0.0	0.50

Рисунок 2.16 – Кінцева база даних

Деякі колонки було видалено, деякі розділено на декілька. Всі пусті значення було заповнено даними.

Висновки до розділу 2

У результаті написання розділу 2 КРМ було проведено опис набору даних. Набір даних містить інформацію про ноутбуки, включаючи їх модель, бренд, технічні характеристики, рік випуску та ціну. Ці дані будуть використовуватися для побудови моделі прогнозування цін на ноутбуки.

Далі було проведено передобробку даних. Передобробкою даних було виявлено деякі проблеми, такі як відсутність даних, дублікати та некоректні значення. За допомогою передобробки було виправлено ці проблеми, включаючи видалення дублікатів та обробку відсутніх значень.

Результатъ опису набору та передобробки даних становлять важливий фундамент для подальших етапів дослідження. Ця інформація допоможе вибрати підходящі методи машинного навчання та побудувати модель прогнозування цін на ноутбуки, що має на меті покращити точність прогнозів.

3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ

3.1 Моделювання програмного забезпечення

Варіант використання фіксує згоду між користувачами системи про її поведінку. Варіант використання описує поведінку системи при її відповідях на запити дійових осіб (actors) в різних умовах. Діаграма варіантів використання [16], зображена на рисунку 3.1, допомагає систематизувати функціональні вимоги та визначити ролі користувачів разом з відповідними правами доступу. Представлено ті типи зв'язків, що мають місце у застосунку, тобто ***generalization***, ***extend*** та ***include relationship***.

Інформацію про акторів представлено у таблиці 3.1.

Таблиця 3.1 – Інформація про акторів

Назва	Опис
Користувач	Має можливість обирати спосіб представлення даних та переглядати результат
Програміст	Аналізує та обробляє дані, навчає модель, візуалізує та прогнозує дані
Адміністратор	Оновлює, змінює, додає та видаляє дані

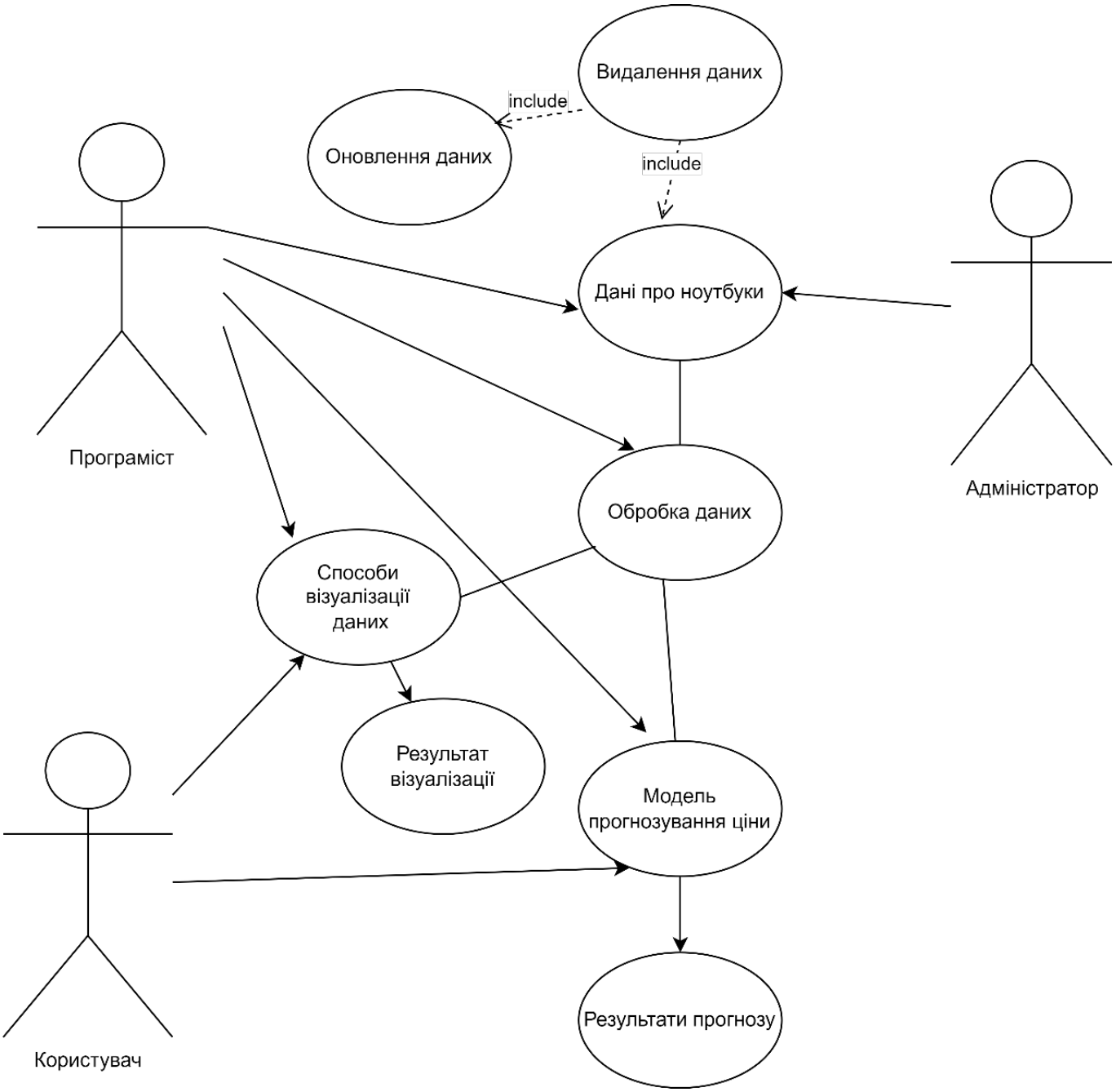

Рисунок 3.1 – Діаграма варіантів використання системи

На діаграмі відображено всі функції системи. Наведена діаграма прецедентів містить можливі сценарії використання для адміністратора та користувача, а також розробника.

В ході підготовки програмного проєкту було створено діаграму, яка зображає можливі дії користувачів, варіанти подальшої поведінки системи. Один з таких прикладів зображено на рис. 3.2.

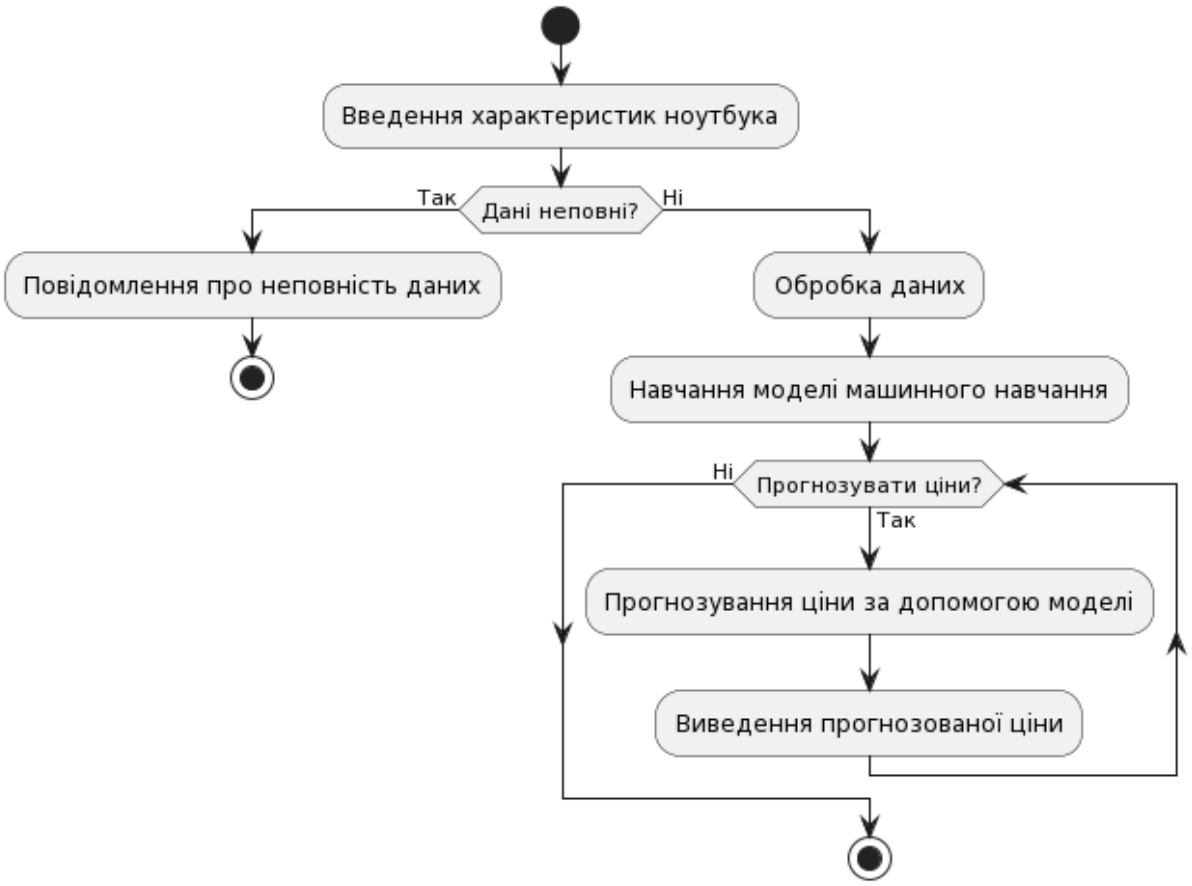

Рисунок 3.2 – Діаграма активності

На діаграмі показано прогнозування за допомогою моделі.

Діаграма пакетів потрібна для організації елементів у групи за будь-якою ознакою із метою спрощення структури та організації роботи із моделлю системи.

Пакет (package) – це інструмент групування, який дозволяє взяти будь-яку конструкцію UML та об’єднати її елементи в одиниці високого рівня. Здебільшого пакети служать для об’єднання класів у групи.

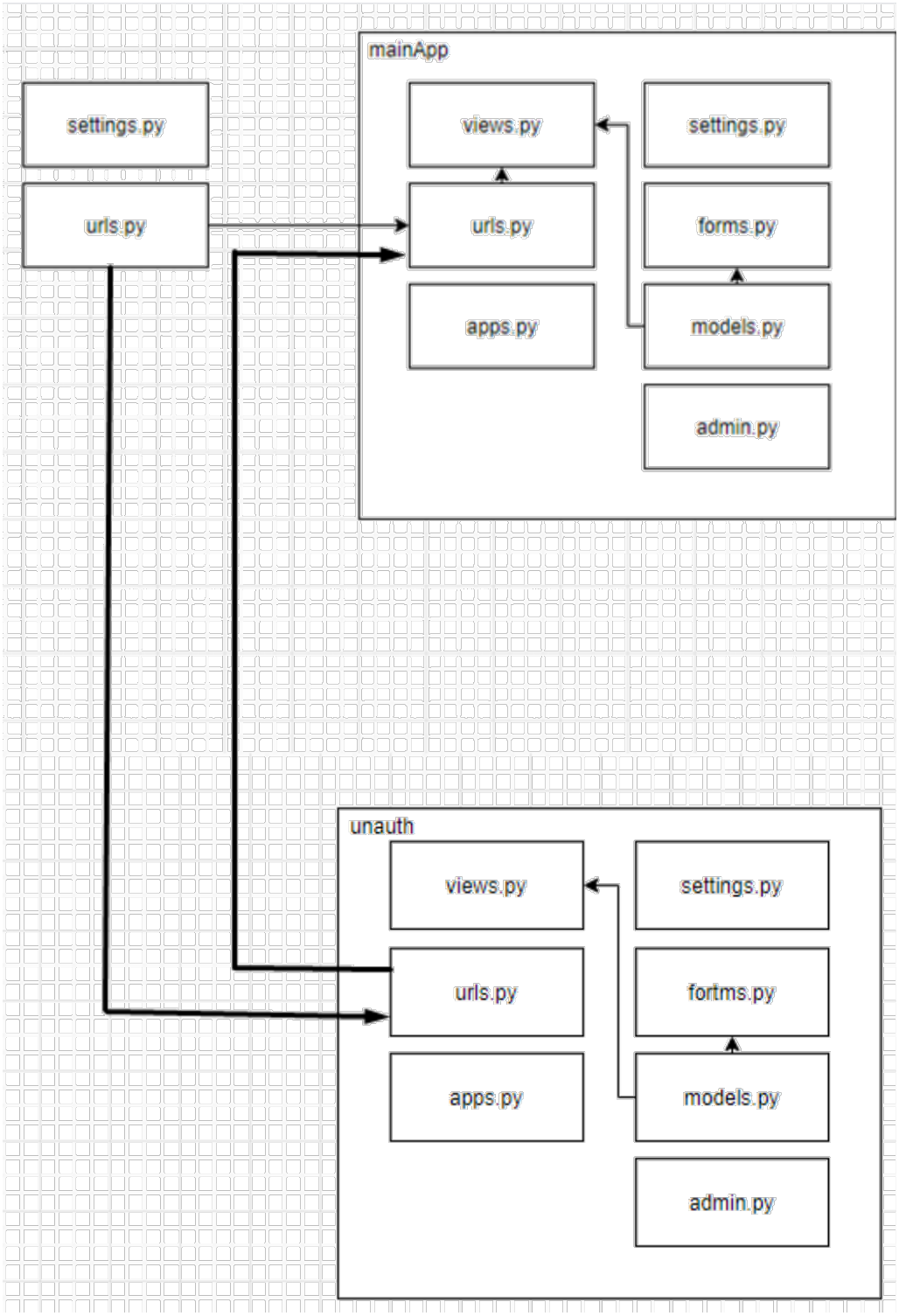

Рисунок 3.3 – Діаграма пакетів

Зображена на рисунку 3.3 діаграма пакетів відображає пакети класів та залежності між зазначеними пакетами.

3.2 Вибір мови програмування

Розробка веб-застосунків вимагає ретельного вибору технологій і підходів. Важливо також зрозуміти, як вони були відібрані для проєкту.

При виборі технології важливо враховувати наступне [18]:

- складність програмного забезпечення;
- наявність ресурсів;
- наявність готових компонентів;
- наявність технічної документації;
- характеристики якості ПЗ;
- витрати на технологію;
- ліцензійну політику;
- вимоги безпеки.

Для прогнозування зазвичай використовуються дві мови програмування: Python та R.

Python – це високорівнева, інтерпретована, мультипарадигмальна мова програмування загального призначення. Вона була розроблена в 1991 році Гвідо ван Россумом і має відкрите джерело коду, що дозволяє вільно використовувати її та розповсюджувати. Також це об’єктно-орієнтована мова програмування із строгою динамічною типізацією даних.

Переваги Python:

1. *Велика кількість бібліотек та інструментів для машинного навчання.*

   Має розгалужену екосистему бібліотек для машинного навчання, таких як NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch, Keras тощо. Ці бібліотеки надають широкий вибір алгоритмів та інструментів для роботи з даними та побудови моделей машинного навчання.

2. *Простий синтаксис. Синтаксис простий та зрозумілий, що полегшує розробку та відлагодження коду. Це робить Python доступним для широкого кола користувачів, навіть для тих, хто не має глибоких знань у програмуванні.*
3. *Активна спільнота. Має велику та активну спільноту користувачів та розробників. Це означає, що завжди можна знайти підтримку, відповіді на питання та розв'язання проблем через форуми, блоги, соціальні мережі та інші ресурси.*
4. *Широке застосування. Використовується у багатьох сферах технологій, включаючи машинне навчання, веб-розробку, аналіз даних, наукові дослідження, автоматизацію тощо. Це робить його універсальним інструментом для різних завдань.*

Недоліки Python:

1. *Швидкодія.* може бути повільнішим через свою інтерпретовану природу. Це може стати проблемою при обробці великих обсягів даних або при вимогливих обчисленнях.
2. *Менша ефективність для паралельної обробки даних.* Може бути менш ефективним для паралельної обробки великих обсягів даних.
3. *Глобальний інтерпретатор (GIL).* В Python існує Global Interpreter Lock (GIL), який унеможливлює виконання потоків Python коду в багатопроцесорних системах. Це може призводити до проблем з ефективністю паралельної обробки даних.

R – це мова програмування та середовище для статистичних обчислень та візуалізації даних. Вона була розроблена у 1993 році й отримала свою назву від імені її розробників - Роберта Гентлмена та Росса Іхаки. Ось детальніше про R:

Переваги R:

1. *Багата статистична інфраструктура.* Велика кількість статистичних пакетів та бібліотек, які доступні для роботи з даними. Ці пакети дозволяють виконувати різноманітні статистичні аналізи, включаючи регресійний аналіз, аналіз варіації, класифікацію та кластеризацію.
2. *Сильна візуалізація даних.* Має потужні інструменти для візуалізації даних, такі як бібліотека ggplot2, яка дозволяє створювати якісні та ефективні графіки для відображення даних.
3. *Активна спільнота користувачів.* Має велику та активну спільноту користувачів та розробників, що робить його ідеальним вибором для тих, хто шукає підтримку, відповіді на питання та рішення проблем.
4. *Широке застосування в науці та дослідженнях.* Широко використовується у наукових дослідженнях, академічних кругах, статистичних аналізах та в інших областях, де потрібно аналізувати великі обсяги даних.
5. *Інтеграція з іншими мовами та інструментами.* Може легко інтегруватися з іншими мовами програмування та інструментами, що дозволяє розробникам використовувати його разом з іншими інструментами та розширювати його функціональність.

Недоліки R:

1. *Швидкодія:* може бути менш швидким через свою інтерпретовану природу та високий рівень абстракції.
2. *Обмежена підтримка паралельних обчислень:* у деяких випадках R може мати обмежену підтримку паралельних обчислень, що може бути проблемою для обробки великих обсягів даних.

3. *Відсутність пакетів для деяких завдань:* деякі спеціалізовані завдання можуть вимагати розробки власних алгоритмів або використання інших мов програмування.
4. *Вивчення кривої:* для новачків в програмуванні R може виявитися відносно складним у вивченні порівняно з іншими мовами програмування.

Отже, і Python, і R мають свої переваги та недоліки. Використовувати потрібно технологію в залежності від поставленого завдання.

Після проведення порівняння та аналізу мов програмування, врахування переваг та недоліків кожної у якості технології розробки програмного забезпечення було обрано Python.

3.3 Вибір технології

Фреймворки Flask і Django - це обидва популярні інструменти для розробки веб-додатків на мові програмування Python. Вони мають різні підходи та філософії, тому їх вибір залежить від конкретних потреб проєкту.

Фреймворк Flask - це легкий, гнучкий та простий у використанні інструмент для розробки веб-додатків на мові програмування Python. Ось кілька ключових характеристик Flask:

Мікрофреймворк. Відноситься до категорії "мікрофреймворків", що означає, що він має мінімальний набір функціоналу, не містить обов'язкових компонентів і завдання розробника - вибирати самому, які інструменти використовувати для свого проєкту.

Легкий у використанні. Пропонує простий та лаконічний синтаксис, що дозволяє розробникам швидко створювати веб-додатки. Його архітектура побудована таким чином, щоб спрощувати процес розробки, забезпечуючи при цьому достатню гнучкість для реалізації різноманітних вимог проєкту.

Розширюваність. хоча Flask має мінімальний функціонал за замовчуванням, він дозволяє легко розширюватися за допомогою різних розширень (extensions) та бібліотек. Це дозволяє використовувати лише ті компоненти, які дійсно потрібні для вашого проєкту, що робить Flask дуже гнучким.

RESTful підтримка. Добре підтримує створення RESTful веб-сервісів, що робить його популярним вибором для розробки API.

Jinja2 для шаблонів. Використовує потужний движок шаблонів Jinja2, що дозволяє легко та ефективно відокремлювати логіку додатку від представлення.

Робочий процес розробки. Має вбудовану підтримку робочого процесу розробки, що дозволяє розробникам швидко створювати, тестувати та розгортати свої веб-додатки.

Адаптований для невеликих проєктів. Через свою простоту та мінімалістичний підхід, Flask часто використовується для розробки невеликих та середніх проєктів, де немає потреби в обширних функціях та компонентах.

Узагальнюючи, Flask - це легкий, елегантний та простий у використанні фреймворк, який надає розробникам гнучкість та контроль над розробкою веб-додатків на Python.

Django - це високорівневий веб-фреймворк на мові програмування Python, розроблений для прискорення розробки веб-додатків шляхом надання готових рішень і стандартизації кращих практик. Ось детальний огляд ключових характеристик Django:

Модульність та вбудовані функції. Містить велику кількість вбудованих функцій і компонентів, таких як система аутентифікації, ORM (Object-Relational Mapping), адміністративна панель, система маршрутизації URL, шаблонізатор, підтримка сесій, міжнародна підтримка, та інші. Це дозволяє розробникам швидко розпочати проєкт і використовувати вже готові рішення.

ORM та бази даних. Постачається з вбудованою ORM, яка дозволяє взаємодіяти з базами даних за допомогою Python-об'єктів, замість прямої роботи з SQL. Він підтримує різні бази даних, такі як PostgreSQL, MySQL, SQLite, та інші.

Адміністративна панель. Надає міцну адміністративну панель, яка дозволяє легко створювати, оновлювати та видаляти дані з бази даних, а також вносити зміни у налаштування адміністратора. Це робить процес управління адміністративними завданнями дуже ефективним.

Масштабованість. Розроблений для масштабованих проєктів. Він має вбудовану підтримку кешування, підтримку розподіленого обробки, а також можливості оптимізації продуктивності, що дозволяє йому ефективно працювати з великими обсягами даних та високим навантаженням.

Шаблонізація. Використовує потужний шаблонізатор, що дозволяє розробникам розділяти логіку додатку від представлення. Він підтримує використання узагальнених шаблонів, що спрощує процес створення та підтримки користувацького інтерфейсу.

Безпека. Надає ряд вбудованих заходів безпеки, таких як захист від CSRF (Cross-Site Request Forgery), захист від SQL-ін'єкцій, вбудована система аутентифікації та авторизації, що дозволяє розробникам створювати безпечні додатки.

Спільнота та документація. Має велику та активну спільноту користувачів, а також добре документовану офіційну документацію, що допомагає розробникам швидко засвоювати та використовувати фреймворк.

Узагальнюючи, Django - це потужний, повнофункціональний та високорівневий фреймворк для розробки веб-додатків на Python, який дозволяє швидко створювати безпечні, масштабовані та гнучкі додатки.

Отже, врахувавши всі переваги та недоліки кожного для розробки програмного забезпечення було орано фреймворк Django.

3.4 Вибір компонентів програмного забезпечення

Для аналізу та прогнозування на мові програмування Python та R можна використовувати різні компоненти програмного забезпечення, які надають широкі можливості для обробки даних, побудови моделей та візуалізації результатів.

Для Python

1. *Бібліотеки для обробки даних та візуалізації:*
   - Pandas для обробки та аналізу даних.
   - NumPy для роботи з масивами та чисельними даними.
   - Matplotlib, Seaborn або Plotly для візуалізації даних.
2. *Бібліотеки для машинного навчання та прогнозування:*
   - Scikit-learn для класичних алгоритмів машинного навчання.
   - TensorFlow або PyTorch для глибокого навчання.
   - XGBoost, LightGBM або CatBoost для градієнтного бустінгу.
3. *Інтерактивні середовища для аналізу даних та виконання коду:*
   - Jupyter Notebook або JupyterLab для створення інтерактивних документів з кодом, текстом та візуалізаціями.
4. *Інструменти для відлагодження та профілювання коду:*
   - Інтегровані функції PyCharm або Visual Studio Code.
   - Інструменти профілювання, такі як cProfile або line_profiler.

Для R

1. *Пакети для обробки даних та візуалізації:*
   - dplyr для маніпуляцій з даними.
   - ggplot2 для створення графіків та візуалізації даних.
   - tidyr для роботи з даними у форматі "tidy data".
2. *Пакети для машинного навчання та прогнозування:*
   - caret для класичних алгоритмів машинного навчання.

- xgboost або lightgbm для градієнтного бустінгу.
- keras або tensorflow для нейронних мереж.

3. *Інтерактивні середовища та інструменти аналізу даних:*
   - RStudio для роботи з кодом R та створення аналітичних звітів.
   - Інтерактивні візуалізаційні бібліотеки, такі як plotly, для створення візуалізацій.
4. *Пакети для відлагодження та профілювання коду:*
   - Інтегровані функції RStudio або пакети, такі як profvis для профілювання коду.

Ці компоненти дозволять вам ефективно виконувати аналіз та прогнозування даних як на мові програмування Python, так і на мові R. Вибір конкретних компонентів залежить від вашого досвіду, потреб проєкту та особливостей даних, з якими ви працюєте.

Висновок до розділу 3

У результаті написання розділу 3 КРМ було створено діаграму варіантів використання для наочної демонстрації можливостей кожного з користувачів.

Далі було розроблено діаграму активності що дає змогу краще зрозуміти функції застосунку.

Також було розроблено діаграму пакетів для кращого розуміння структури проєкту.

Для проєктування було використано мову моделювання UML. Для створення діаграм знадобилося таке програмне забезпечення, як Lucidchart [15] та Software Ideas Modeler.

Далі було обґрунтовано вибір технологій для реалізації системи. З'ясовано, які фактори впливають на вибір технологій. У якості мови програмування було обрано

Pyhon, а у якості середовища розробки було обрано PyCharm. Також було обрано додатково фреймворк Django та бібліотеки такі, як:

1. Pandas для обробки та аналізу даних,
2. NumPy для роботи з масивами та чисельними даними.
3. Matplotlib, Seaborn та Plotly для візуалізації даних.
4. Scikit-learn для класичних алгоритмів машинного навчання.
5. XGBoost для градієнтного бустінгу.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Опис програмної реалізації

Підготовка до розробки програмного забезпечення

Перед тим, як почати розробку застосунку, потрібно встановити середовище розробки. PyCharm має декілька версій: Community та Professional. Було обрано версію Community, оскільки вона є безкоштовною для студентів. Завантажується із офіційного сайту компанії JetBrains, яка є її розробником. Після встановлення потрібно додатково завантажити фреймворк Django, який входить до розширень середовища розробки. Також при необхідності можна обрати інші компоненти. Перевагою є те, що пізніше можна дозавантажити потрібні компоненти або навпаки видалити їх. На рисунку 4.1 показана сторінка встановлення додаткових компонентів.

Створення проєкту

Середовище розробки PyCharm Community Edition підтримує фреймворк Django й працює з різними версіями Python. Для створення проєкту в PyCharm необхідно обрати Create New Project (рисунок 4.1).

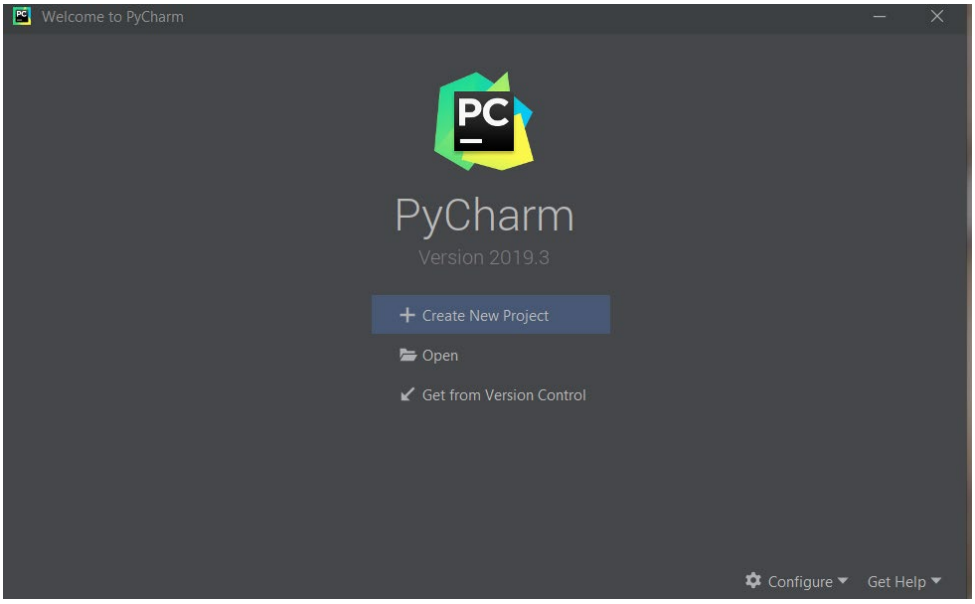

Рисунок 4.1 – Створення проєкту в PyCharm

Далі у вікні створення нового проєкту необхідно обрати Django, здійснити необхідні налаштування проєкту і натиснути Create (рисунок 4.2). Після цього чекаємо скачування фреймворку та створення проєкту. Проектом називається сукупність усього програмного коду, що становить розроблювальний сайт. Фізично він є папкою, у якій перебувають різноманітні файли з вихідним кодом (рисунок 4.3).

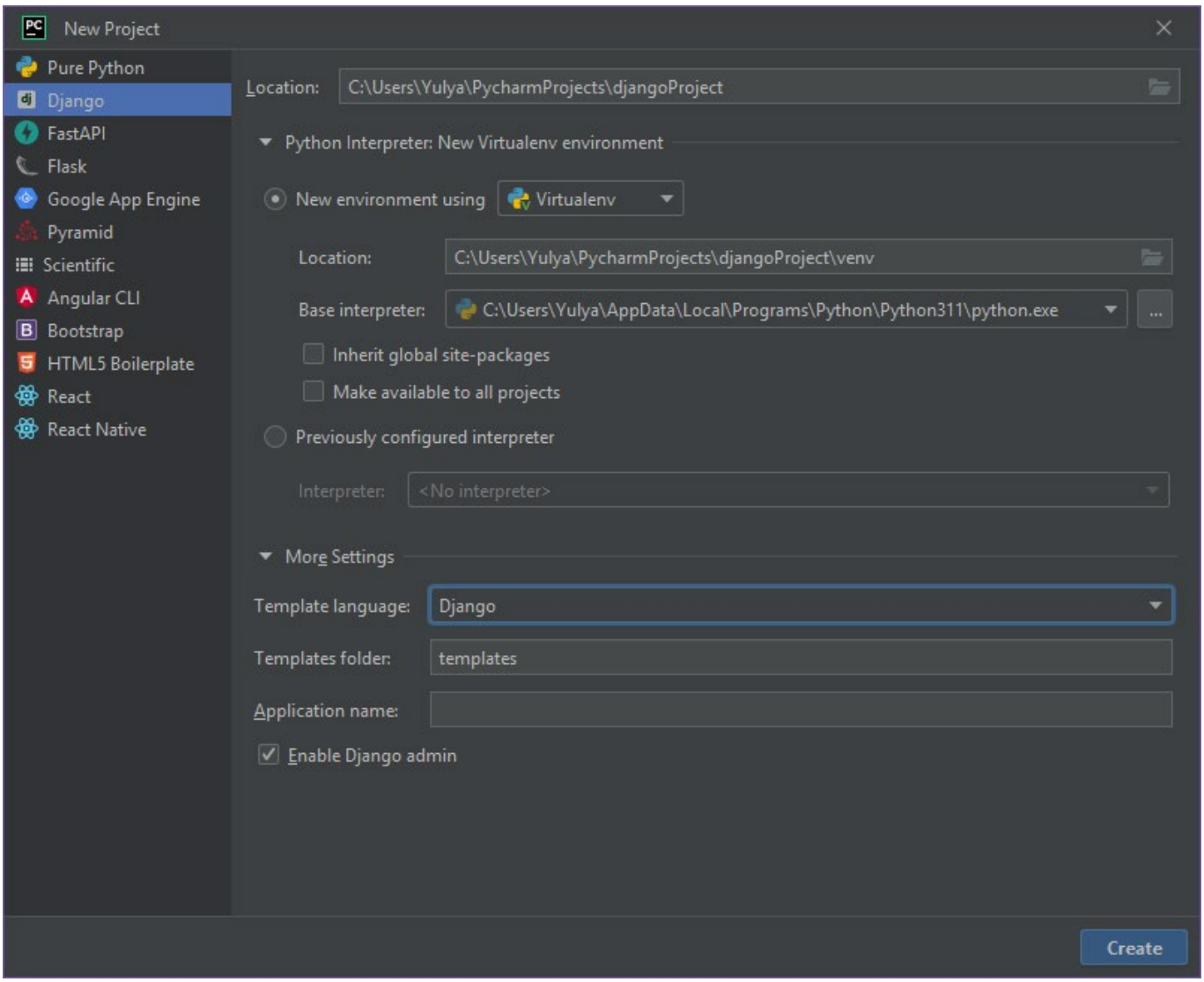

Рисунок 4.2 – Створення нового проєкту Django

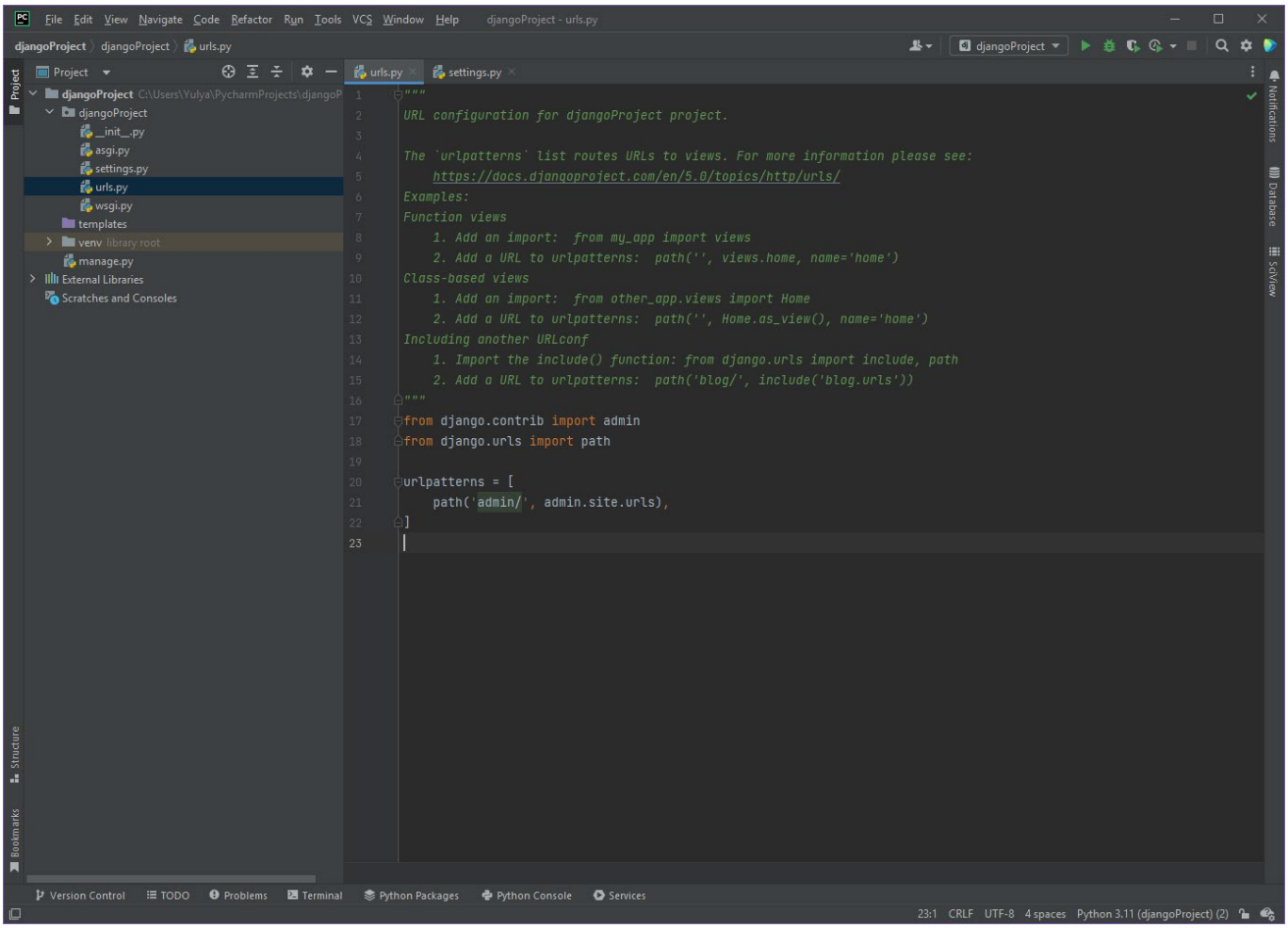

Рисунок 4.3 – Структура проєкту Django

У папці із проєктом буде створена структура файлів. Розглянемо їх докладніше.

manage.py – програмний файл із кодом однойменної утиліти, з використанням якої виробляються різні дії над самим проєктом.

Внутрішня папка ***projectDjango*** – формує пакет мови Python, що містить модулі, які відносяться до проєкту і задають його конфігурацію (ключові настроювання). Назва цього пакету збігається з назвою проєкту. У даному пакеті лежать файли:

__init.py__ – порожній файл, що повідомляє Python, що папка, у якій він перебуває, є повноцінним пакетом.

settings.py – модуль із настроюваннями самого проєкту. Включає опис конфігурації бази даних проєкту, шляхи ключових папок, важливі параметри, пов'язані з безпекою.

urls.py – модуль із маршрутами рівня проєкту.

wsgi.py – модуль, що зв'язує проєкт із веб-сервером. Використається при публікації готового сайту в Інтернеті.

asgi.py – модуль призначений для забезпечення стандартного інтерфейсу між асинхронними веб-серверами Python, фреймворками й додатками.

Структура проєкту Django схожа на шаблон MVC, але дещо відрізняється: Model-View-Template (MVT). Структуру представлено на рисунку 4.4.

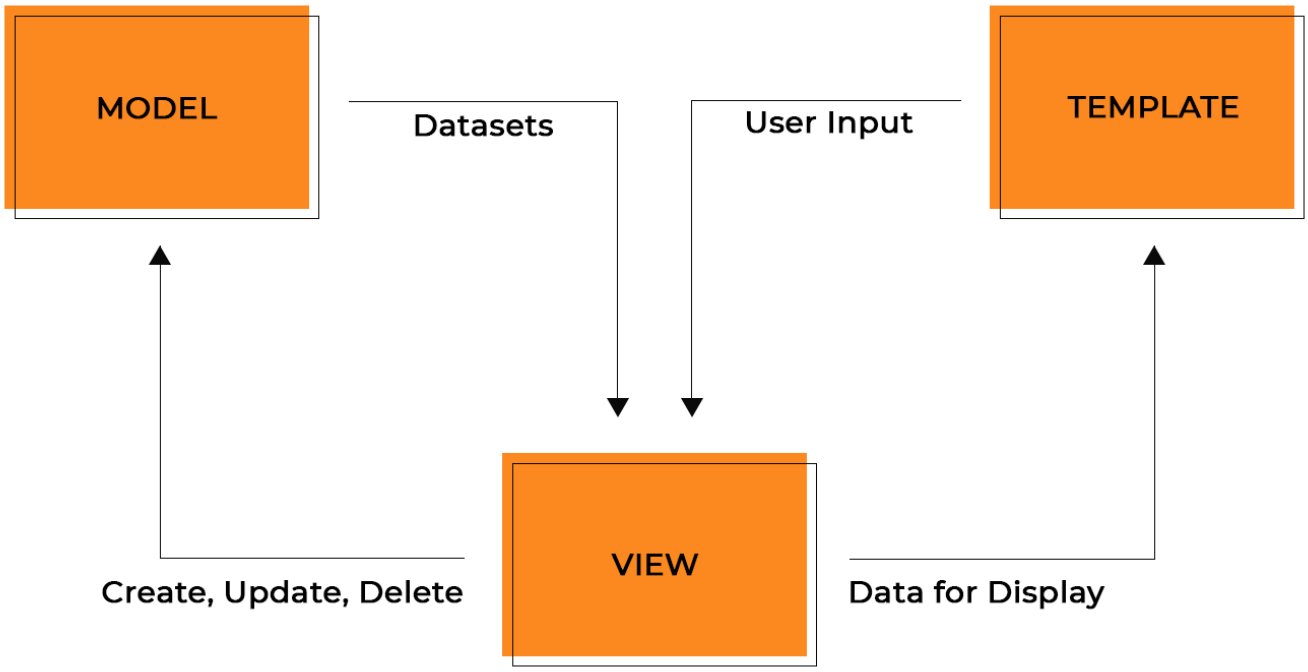

Рисунок 4.4 – Структура проєкту Django

Веб-застосунки, написані на Django, звичайно групують код в окремі файли.

URLs: хоча можна обробляти запити з кожної URL-адреси за допомогою однієї функції, набагато зручніше писати окрему функцію для обробки кожного ресурсу. URL-маршрутизатор використовується для перенапрямку HTTP-запитів у відповідне

подання на основі URL-адреси запиту. Крім того, URL-маршрутизатор може витягати дані з URL-адреси відповідно до заданого шаблону й передавати їх у відповідну функцію відображення (view) у вигляді аргументів.

View: View (англ. "відображення") – це функція оброблювача запитів, що одержує HTTP-запити й повертає відповіді. Функція view має доступ до даних, необхідним для задоволення запитів, і делегує відповіді в шаблони через моделі.

***Models*:** моделі є об’єктами Python, які визначають структуру застосунку й надають механізми для керування (додавання, зміни, видалення) і виконання запитів у базу даних.

Templates: Template (англ. "шаблон") – це текстовий файл, що визначає структуру або розмітку сторінки (наприклад HTML-сторінки), з полями для підстановки, які використаються для висновку актуального вмісту. View може динамічно створювати HTML-сторінки, використовуючи HTML-шаблони й заповнюючи їхніми даними з моделі (model). Шаблон може бути використаний для визначення структури файлів будь-яких типів, не обов'язково HTML.

Також було додатково створено папку ***static– > images,*** у якій зберігаються картинки, які були створені під час візуалізації даних.

Також додатково були створенні файли ***forms.py*** – клас для взаємодії із даними бази даних та ***prediction.py*** – основний код застосунку із усіма функціями.

Після цього створюємо необхідні додатки, які і будуть складовими нашого веб-застосунку.

4.2 Первинний візуальний аналіз даних

Методи первинного візуального аналізу даних

Після завершення первинного аналізу даних можна перейти до первинного візуального аналізу даних. Цей етап дозволяє отримати загальне уявлення про розподіл змінних, виявити взаємозв'язки між змінними та ідентифікувати потенційні аномалії або виключні значення.

Під час первинного візуального аналізу даних використовуємо графіки та візуалізації для отримання загального уявлення про розподіл та взаємозв'язки між змінними в наборі даних. Це дозволяє нам швидко виявити патерни, аномалії та особливості даних.

Візуалізація даних. Створити графіки та діаграми для візуального представлення даних. Наприклад, гістограми для числових змінних, графіки розсіювання для аналізу залежностей між змінними, кругові діаграми для категоріальних змінних. Візуалізація даних допоможе виявити закономірності, кореляції та потенційні викиди або аномалії.

Кореляційний аналіз. Використання кореляційної матриці або графіків розсіювання для виявлення зв'язків між числовими змінними. Це дозволяє з'ясувати, які змінні можуть бути взаємозалежними та як вони впливають одна на одну. Кореляційний аналіз може вказати на потенційні фактори, які слід враховувати при подальшому моделюванні або аналізі даних.

Розуміння категоріальних змінних. Для категоріальних змінних вивчити розподіл категорій та їх частоту. Дізнатися, які категорії є найпоширенішими та чи є які-небудь незвичайні або рідкісні категорії, які можуть вплинути на аналіз.

Засоби візуалізації даних

Гістограми. Вони допомагають визначити розподіл змінних. Наприклад, можна побудувати гістограму для брендів ноутбуків, щоб побачити, як бренди розподілені по діапазону значень.

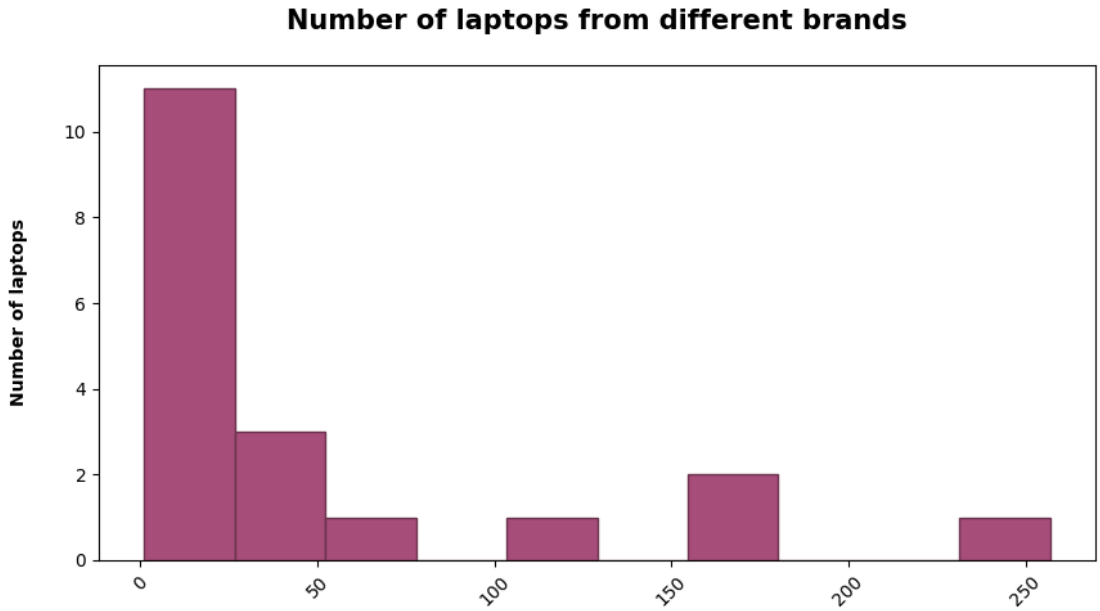

Рисунок 4.5 – Розподіл брендів

Діаграми розсіювання. Ці діаграми відображають залежність між двома змінними. Наприклад, можна побудувати діаграму розсіювання для операційної системи та ціни ноутбуків, щоб побачити, як вони взаємозв'язані.

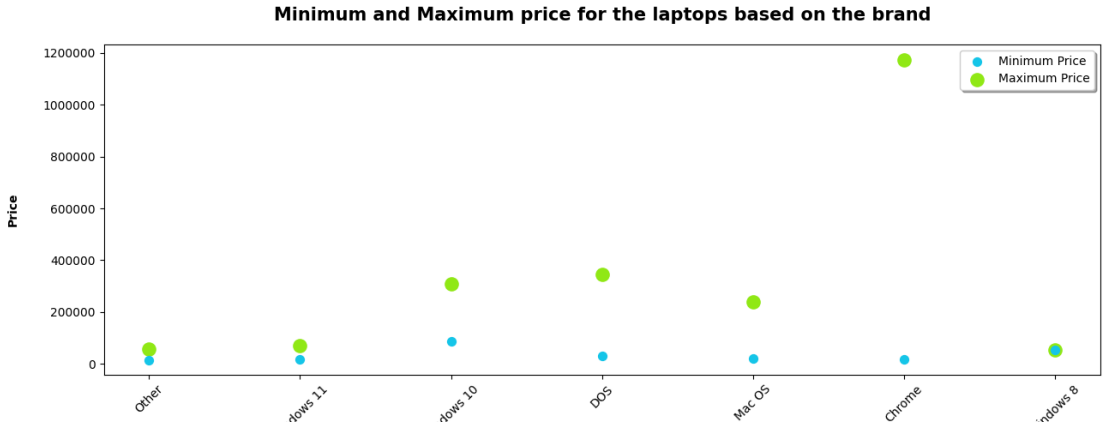

Рисунок 4.6 – Взаємозв’язок операційної системи та ціни

Коробкові діаграми. Вони використовуються для візуалізації розподілу змінних та виявлення викидів або аномалій. Коробкова діаграма [4] може показати мінімальне значення, перший квартиль, медіану, третій квартиль та максимальне значення змінної.

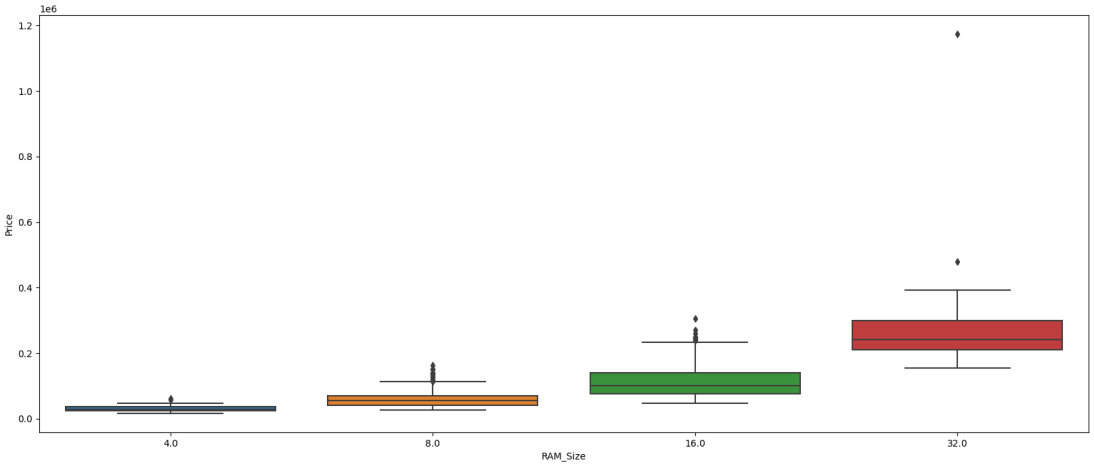

Рисунок 4.7– RAM_Size

Стовпчикові діаграми. Вони дозволяють порівняти значення змінних між категоріями. Наприклад, можна побудувати стовпчикову діаграму, щоб порівняти середню ціну ноутбуків за брендами.

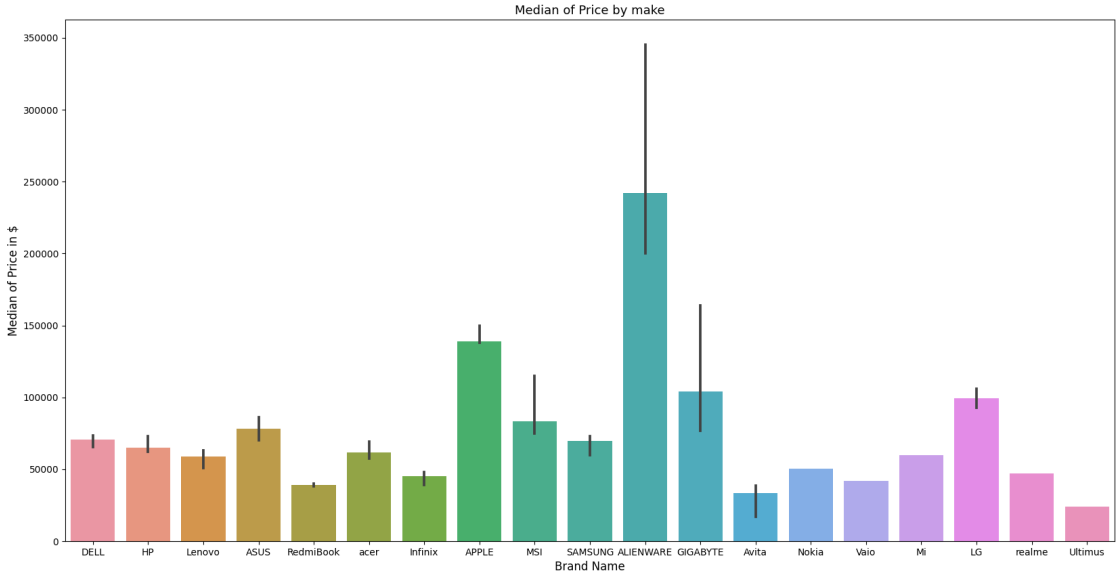

Рисунок 4.8 – Середня ціна ноутбуків за брендами

Кругові діаграми. Вони використовуються для візуалізації відносної частки або пропорцій різних категорій. Наприклад, можна побудувати кругову діаграму, щоб показати відсоткове співвідношення процесорів ноутбуків у представленому наборі даних.

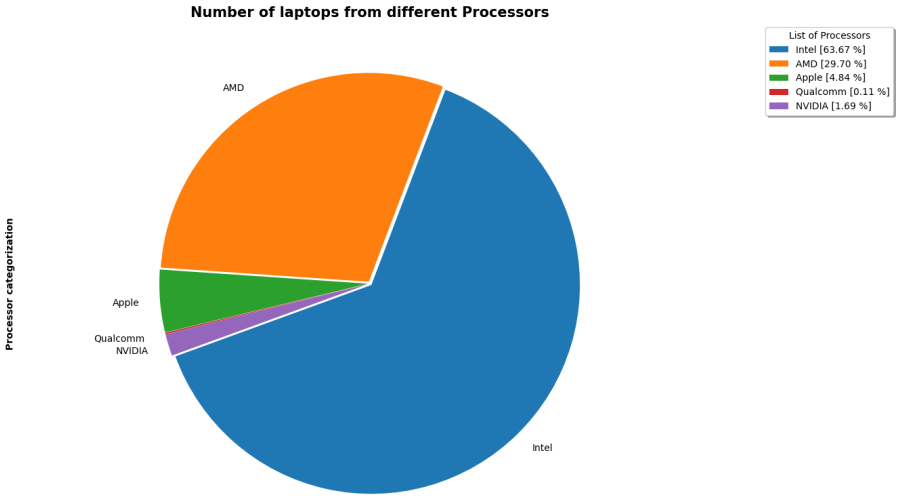

Рисунок 4.9 – Співвідношення процесорів ноутбуків

Графік розподілу. Використовується для відображення розподілу числової змінної шляхом накладання графіка її ймовірності на осі x.

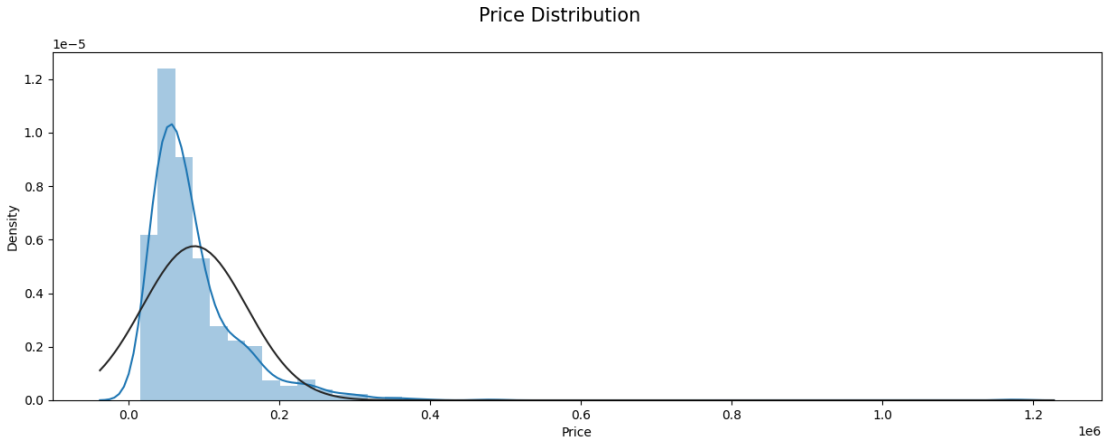

Рисунок 4.10 – Розподіл ціни

Лінійні графіки. Вони використовуються для відображення залежності змінної від часу або іншої змінної. Наприклад, ви можна побудувати лінійний графік, щоб відслідковувати зміну середньої ціни ноутбуків протягом року.

Теплові карти. Вони використовуються для візуалізації матричних даних, де кожному елементу матриці відповідає колірна шкала. Наприклад, ви можна побудувати теплову карту, щоб відобразити кореляцію між різними характеристиками ноутбуків.

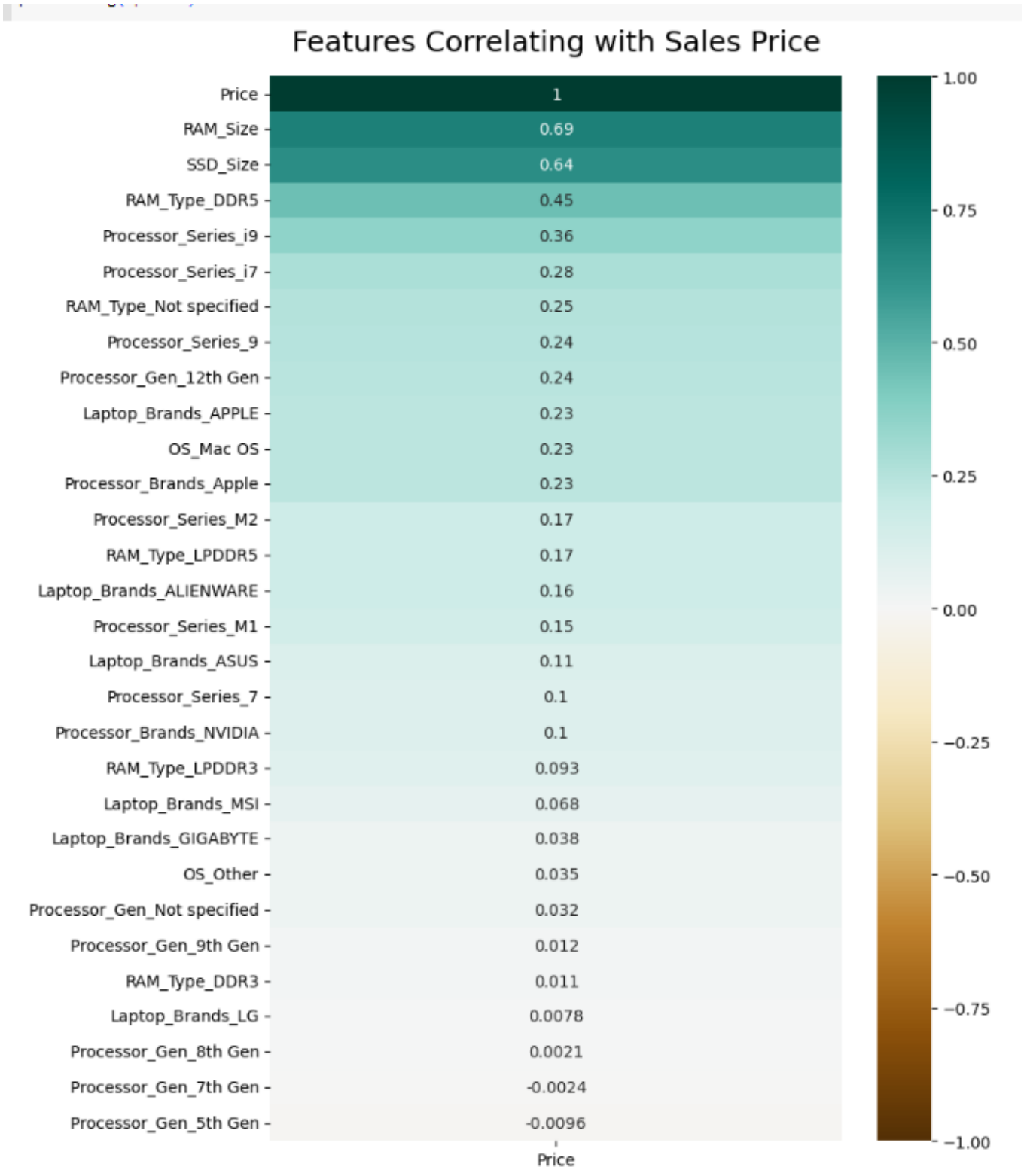

Рисунок 4.11 – Кореляція між характеристиками ноутбуків

Ці методи візуального аналізу даних допоможуть отримати загальне уявлення про розподіл, взаємозв'язки та особливості даних у наборі даних про ноутбуки. Вони дозволяють виявити тенденції, аномалії та патерни, що можуть бути корисними при подальшому аналізі та моделюванні даних.

4.3 Розробка функціоналу

Перед початком кодування потрібно встановити та імпортувати всі необхідні бібліотеки. Встановлення та імпорт виконується в окремому файлі окремо, тобто лише потрібних бібліотек для певного коду. У наступному коді наведений приклад.

```
import base64

import numpy as np
import pandas as pd
from django.shortcuts import render
from pandas.core.interchange import buffer
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import Lasso
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import Ridge
from sklearn.model_selection import cross_val_score, RandomizedSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error
import time
```

Для зв’язку з клієнтською частиною потрібні представлення (views) або доступ до ресурсів, які надає сервер які приймають http-запити від веб-клієнтів та повертають

http-відповіді. Вони описуються у файлі views.py. Щоб підключити представлення, необхідно додати URL до файлу urls.py. Покроково створюється модель, для неї представлення, яке конфігурується потрібною аутентифікацією та видом запитів через серріалізатори. У результаті є ресурси, які використовуютсья клієнтом через

POST, GET ,PUT, DELETE запити і отримують результати у JSON форматі.

Модель для взаємодії із даними.

```
# models.py in yourappname
from django.db import models


class Laptop(models.Model):
    title = models.CharField(max_length=100)
    laptop_brand = models.CharField(max_length=100)
    processor_brand = models.CharField(max_length=100)
    processor_series = models.CharField(max_length=100)
    processor_gen = models.CharField(max_length=100)
    ram_size = models.IntegerField()
    ram_type = models.CharField(max_length=100)
    hdd_size = models.IntegerField()
    ssd_size = models.IntegerField()
    os = models.CharField(max_length=100)
    display = models.CharField(max_length=100)
    price = models.FloatField()

    # Define other fields based on your data

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

```
    class Meta:
        app_label = 'AppTestKRM'
```

Функція для візуалізації даних на сторінці.

```
def generate_visualization(request):
    if request.method == 'POST':
        form = LaptopForm(request.POST)
        if form.is_valid():
            new_laptop_instance = form.save(commit=False)
            new_laptop_instance.save()

            pd.DataFrame([{
                'Laptop_Brands': new_laptop_instance.laptop_brand,
                'Processor_Brands': new_laptop_instance.processor,
                'Processor_Series':
new_laptop_instance.processor_series,
                'Processor_Gen': new_laptop_instance.processor_gen,
                'RAM_Size': new_laptop_instance.ram_size,
                'RAM_Type': new_laptop_instance.ram_type,
                'HDD_Size': new_laptop_instance.hdd_size,
                'SSD_Size': new_laptop_instance.ssd_size,
                'OS': new_laptop_instance.os,
                'Display': new_laptop_instance.display,
                'Price': new_laptop_instance.price
            }])

            # Continue with your plot generation or redirection
            image_png = base64.b64encode(buffer.read()).decode('utf-8')
            return render(request, 'AppTestKRM/visualization.html',
{'image_png': image_png})
        else:
            form = LaptopForm()
```

```
        return render(request, 'AppTestKRM/visualization.html',
{'form': form})
```

У наступному коді наведено функцію лінійної регресії.

```
def run_linear_regression(X_train, y_train, X_test, y_test):
    # Define categorical and numerical columns
    cat_cols = ['OS', 'Laptop_Brands', 'Processor_Brands',
                'Processor_Series', 'Processor_Gen',
                'RAM_Type', 'Display'
                ]
    num_cols = ['RAM_Size', 'HDD_Size', 'SSD_Size']

    # Create the pipeline steps
    num_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='mean')),
        ('scaler', StandardScaler())
    ])

    cat_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('encoder', OneHotEncoder(drop='first', sparse_output=False,
handle_unknown='ignore'))
    ])
    step1 = ColumnTransformer(
        transformers=[
            ('num', num_transformer, num_cols),
            ('cat', cat_transformer, cat_cols)
        ]
    )

    step2 = LinearRegression()
    # Combine the steps into a pipeline
```

```
    pipe = Pipeline([('step_1', step1), ('step_2', step2)])

    # Fit the pipeline
    pipe.fit(X_train, y_train)

    # Measure execution time
    start_time = time.time()
    prediction_result = pipe.predict(X_test)
    end_time = time.time()
    execution_time = end_time - start_time

    # Calculate evaluation metrics
    r2 = round(r2_score(y_test, prediction_result), 5)
    mae = round(mean_absolute_error(y_test, prediction_result), 5)

    mse = round(mean_absolute_error(y_test, prediction_result), 5)
    execution_time = round(execution_time, 5)

    # Return the fitted pipeline
    return {
        "Prediction_result": prediction_result[0],
        "MSE": mse,
        "MAE": mae,
        "R2_Score": r2,
        "Execution_time": execution_time
    }
```

Налаштування шляхів для відображення сторінок.

```
from django.conf import settings
from django.conf.urls.static import static
from django.contrib import admin
from django.urls import path
```

```
from AppTestKRM.views import predict_price, result_view, interface_view

urlpatterns = [
    path('admin/', admin.site.urls),

    path('result_view/', result_view, name='result-page'),

    path('predict/', predict_price, name='predict-price'),

    path('interface/', interface_view, name='interface_view'),

    path('', interface_view, name='home'),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

Після створення сервера з базою даних, від якої надані посилання керування ресурсами, описують відображення цих ресурсів користувачеві через клієнтський застосунок, реалізований з допомогою HTML та CSS.

Користувачеві пропонується два варіанти використання застосунку: візуалізація даних та прогнозування. Оскільки все виконується на одній сторінці, то при використанні одного блоку, інший прихований. Для цього створено окрему JavaScript функцію.

```
      <script>
    // Show/hide choices based on the selected method
    document.getElementById('method_choice').addEventListener('change',
function () {
        let visualizationChoices =
document.getElementById('visualization_choices');
        let predictionChoices =
document.getElementById('prediction_choices');
        let CrossValidationScores =
document.getElementById('Cross_Validation_Scores');
```

```
        if (this.value === 'visualization') {
            visualizationChoices.style.display = 'block';
            predictionChoices.style.display = 'none';
            CrossValidationScores.style.display = 'none';
        } else if (this.value === 'prediction') {
            visualizationChoices.style.display = 'none';
            predictionChoices.style.display = 'block';
            let predictionMethod =
document.getElementById('prediction_method').value;
            if (predictionMethod === 'ridge') {
                CrossValidationScores.style.display = 'block';
            } else {
                CrossValidationScores.style.display = 'none';
            }
        } else {
            visualizationChoices.style.display = 'none';
            predictionChoices.style.display = 'block';
            CrossValidationScores.style.display = 'none';
        }
    });
</script>
```

Також для розділення виводу результатів використовується блок коду ***if-else.***

Код блоку наведено нижче.

```
{% if method %}
    <h2>Result for {{ method }}:</h2>
    {% if result %}
        {% if method|lower in visualization_methods %}
            <img src="{{image_url }}" alt="{{ method }}">
        {% else %}
            <p>RESULT {{ result }}</p>
            <p>Prediction result: {{ result.Prediction_result }}</p>
```

```
            <p>MSE: {{ result.MSE }}</p>
            <p>MAE: {{result.MAE }}</p>
            <p> R2 Score: {{ result.R2_Score}}</p>
            <p>Execution Time: {{ result.Execution_time }} seconds</p>
            <p>PRED: {{ result.pred }}</p>
            <p id="Cross_Validation_Scores">Cross_Validation_Scores {{
result.Cross_Validation_Scores }}</p>
            <p>Best_Ridge_Estimator {{ result.Best_Ridge_Estimator
}}</p>
            <p>Test_Score {{ result.Test_Score }}</p>
            <p>Estimator {{ result.Estimator }}</p>
        {% endif %}
    {% endif %}
{% endif %}
```

Також для відображення інформації використовуються змінні із файлу ***views.py*** (представлення). Для цього використовується такий фрагмент коду:

```
      <p> result {{ result }}</p>
<p>method {{ method }}</p>
<p>url {{ image_url }}</p>
<p>png {{ image_png }}</p>
<p>visualization_methods {{ visualization_methods }}</p>
<p> R2 Score: {{ r2_score }}</p>
<p>MAE: {{result.mae }}</p>
<p>Prediction result: {{ result }}</p>
<p>Execution Time: {{ time_taken }} seconds</p>
```

Код повної сторінки для відображення даних наведено у додатку А.

4.3 Інтерфейс користувача

Розроблений вебзастосунок прогнозування цін ноутбуків складається із однієї сторінки в залежності від вибору користувача. Спочатку користувач потрапляє на сторінку вибору одного із методів: візуалізувати дані чи прогнозувати ціни (рисунок 4.12).

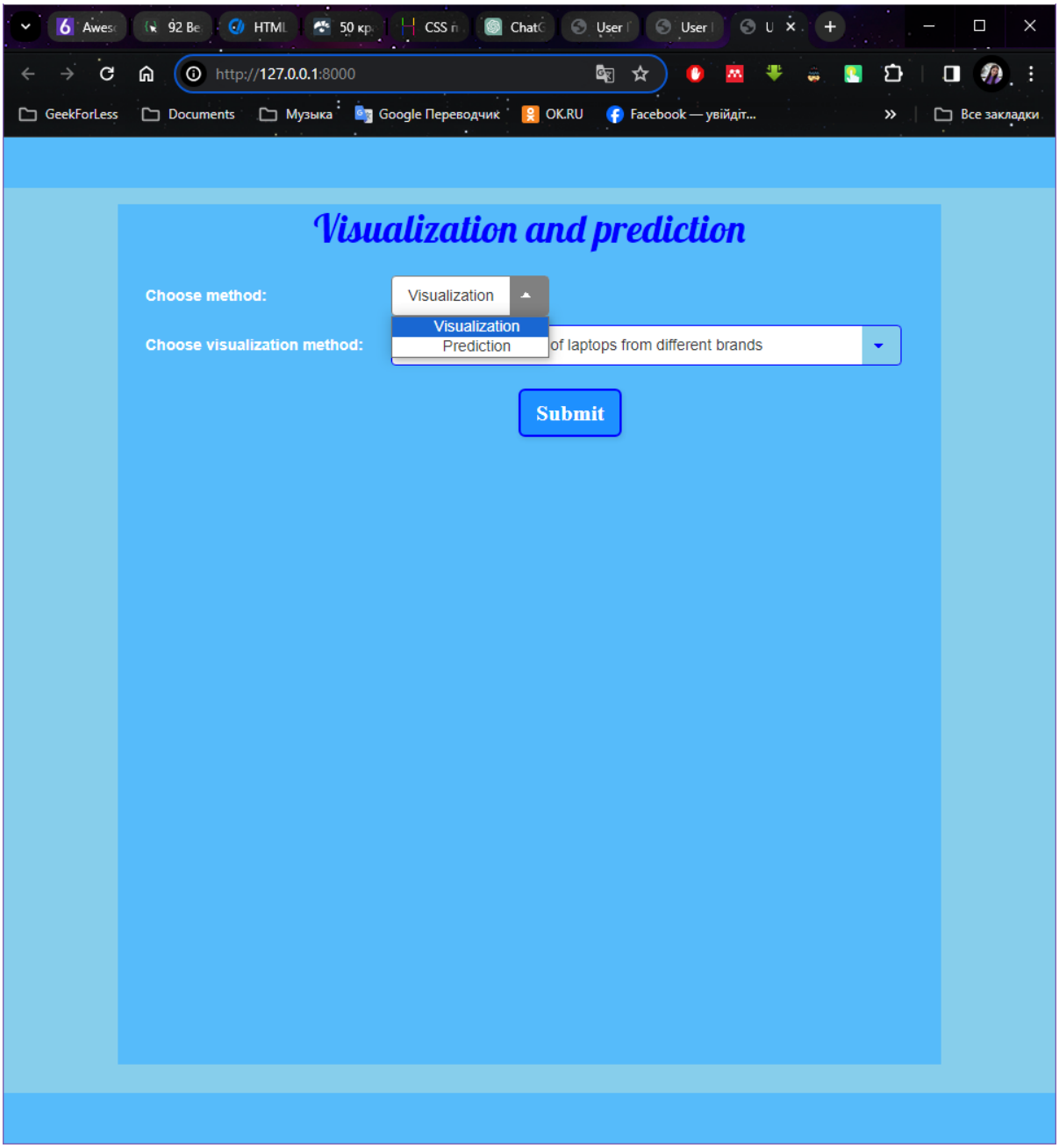

Рисунок 4.12 – Вигляд початкової сторінки

Після цього сторінка оновлюється. Якщо обрано метод візуалізації даних, то відображається певний спосіб візуалізації даних.

На рисунку 4.13 показано вибір способу візуалізації.

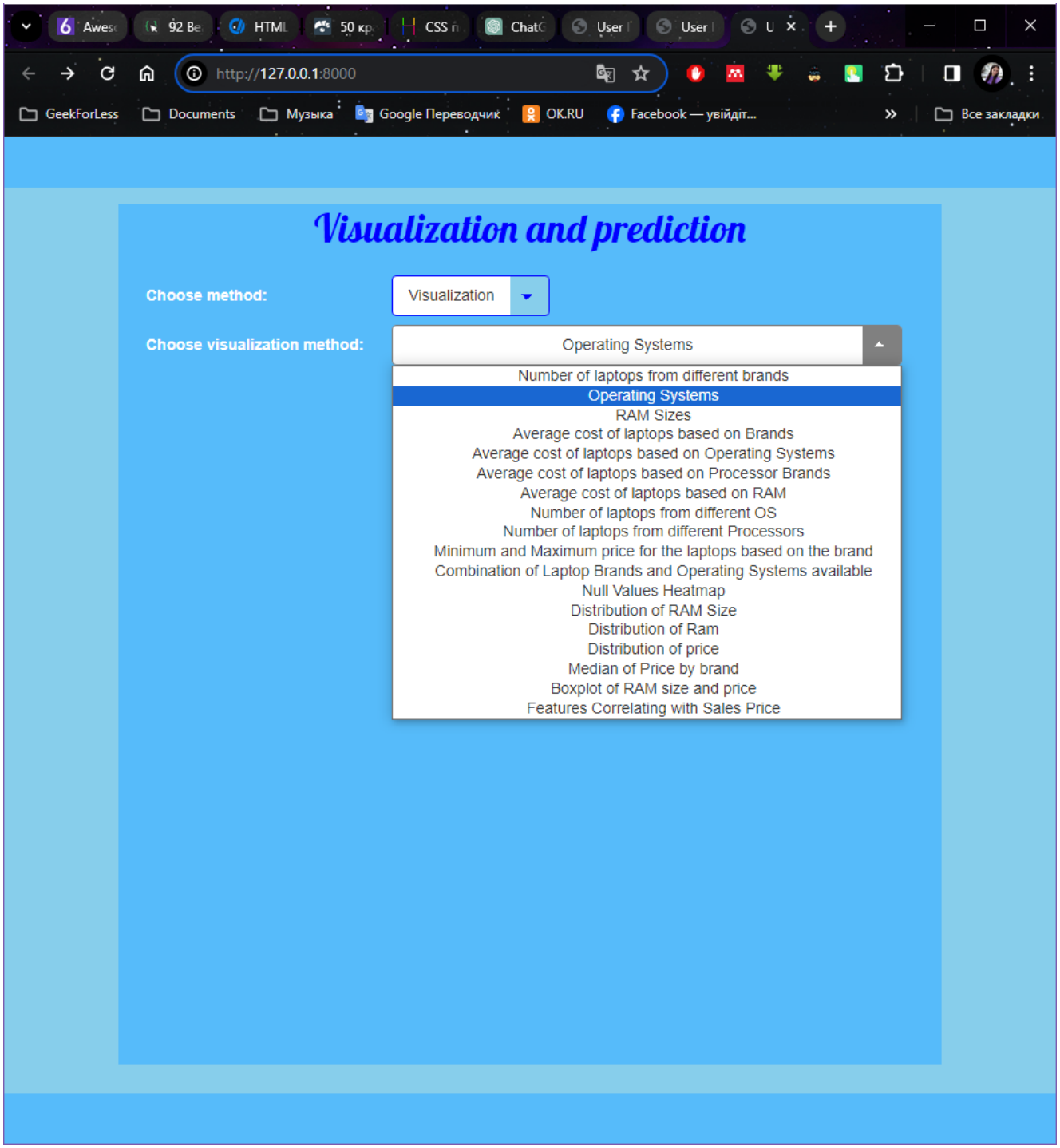

Рисунок 4.13 – Вибір способу візуалізації

На рисунках 4.14 – 4.17 представлено різні методи візуалізації.

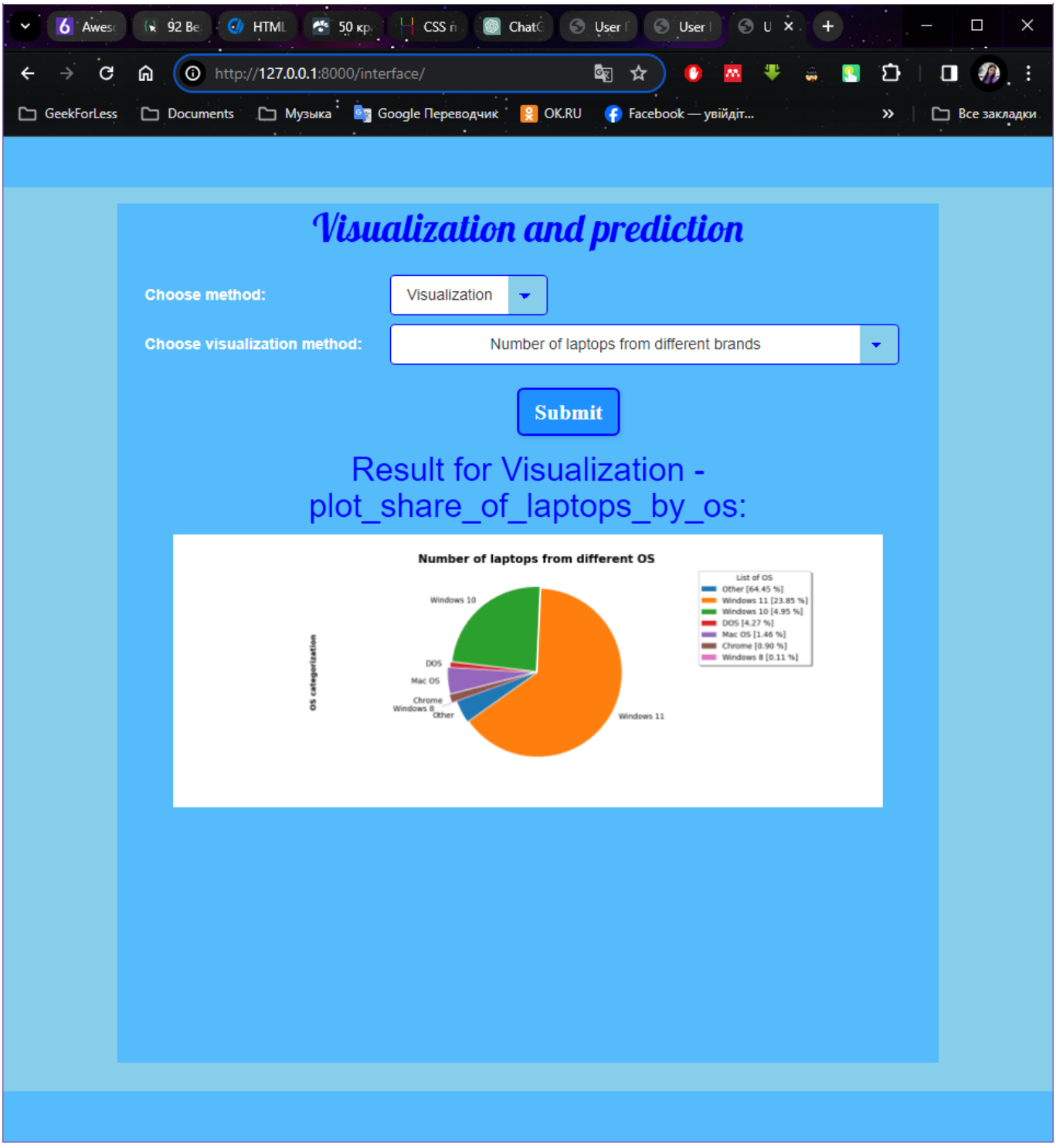

Рисунок 4.16 – Кількість ноутбуків в залежності від операційної системи

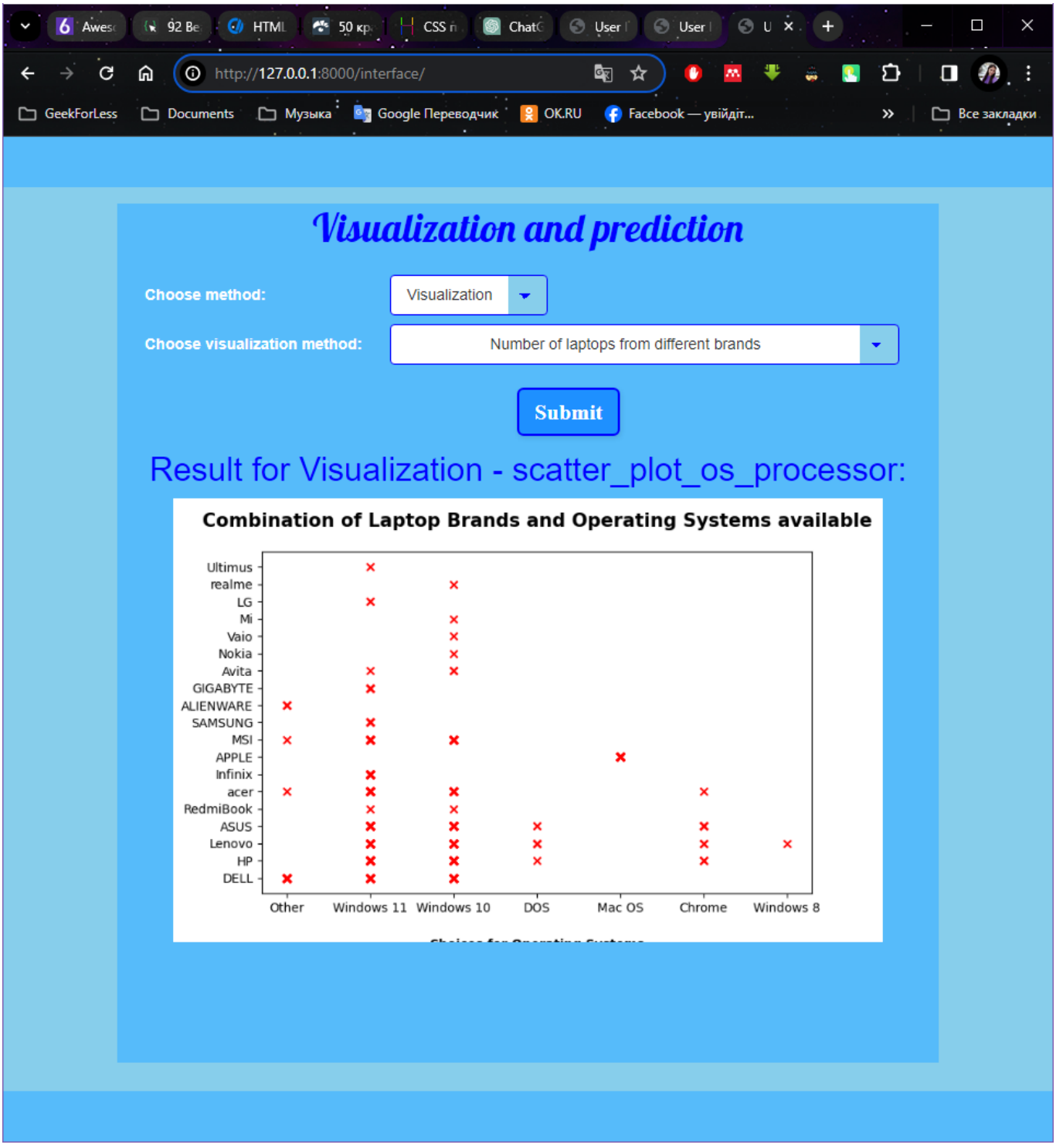

Рисунок 4.17 – Наявність операційної системи для певного бренду

На рисунку 4.18 показано вибір способу прогнозування ціни.

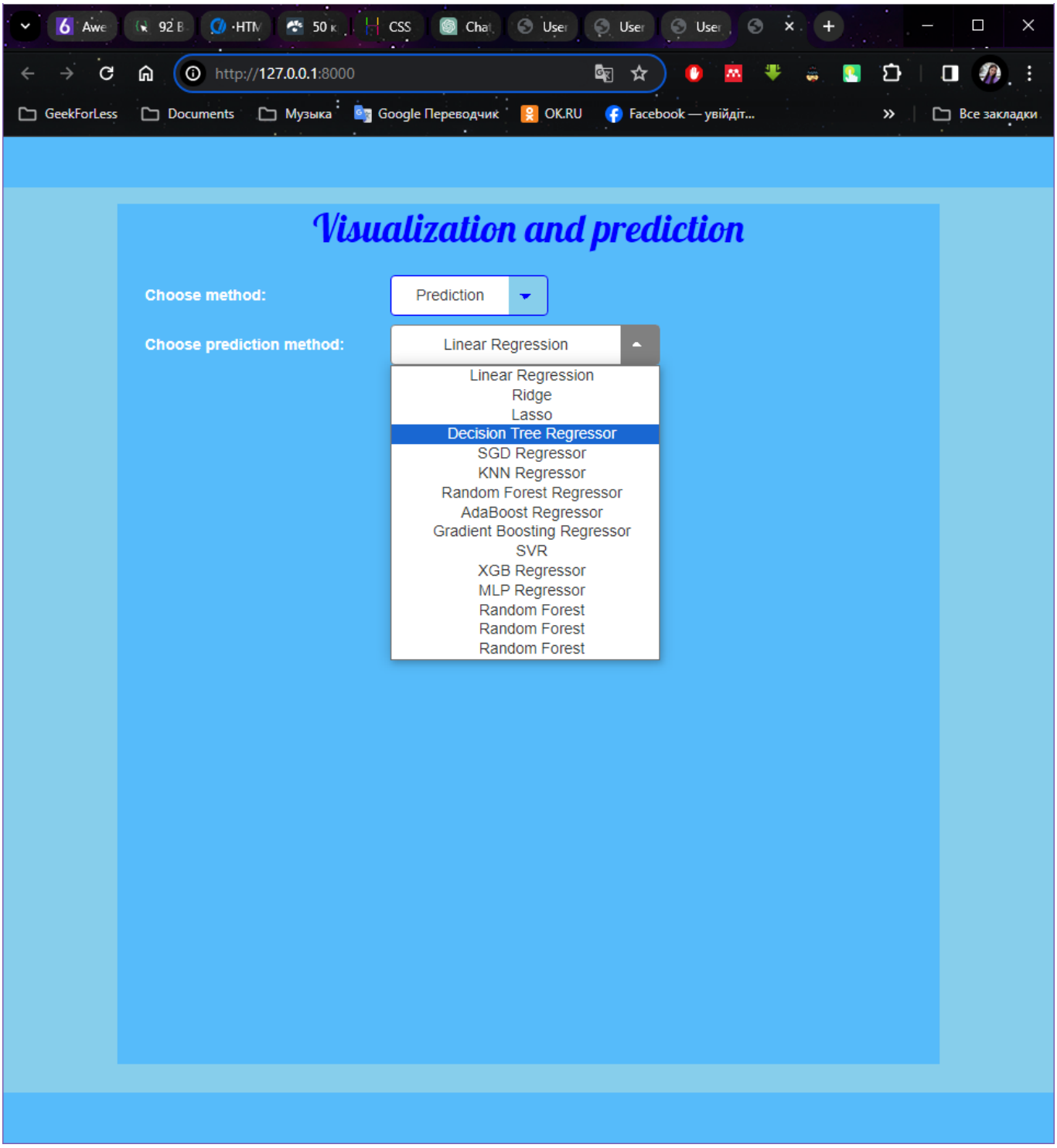

Рисунок 4.18 – Вибір способу прогнозування

Якщо обрано метод прогнозування цін, тоді відображається результат певного методу прогнозування (рисунок 4.19).

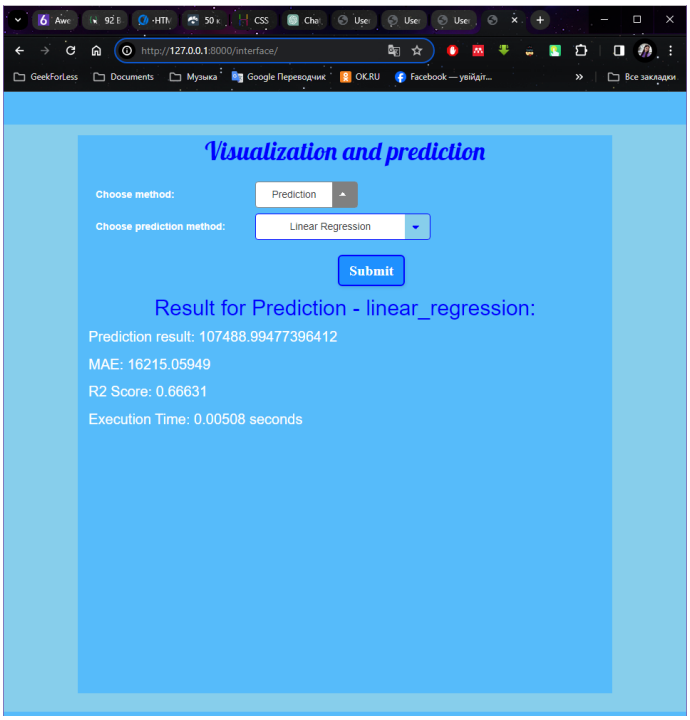

Рисунок 4.19 – Прогнозування методом лінійної регресії

На даному рисунку наведено приклад прогнозування ціни за допомогою методу лінійної регресії.

Висновки до розділу 4

У результаті написання розділу 4 було описано створення проєкту та його структуру, всі його файли та їх короткий опис.

У наступному етапі було проведено первинний візуальний аналіз даних, який включав гістограми цін на ноутбуки та діаграму розсіювання технічних характеристик та цін. Аналіз показав, що розподіл цін має схожість до нормального розподілу з деякими викидами, і що існує залежність між деякими технічними характеристиками та цінами на ноутбуки.

Також було продемонстровано роботу застосунку, наведено інтерфейс програми та всі її функції.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра було розроблено вебзастосунок прогнозування цін ноутбуків із використанням методів машинного навчання. Даний проєкт надає можливість візуалізувати дані та прогнозувати ціни ноутбуків.

Для досягнення мети було виконано **наступні завдання:**

1) Зібрано дані для аналізу.
2) Проведено описовий аналіз даних.
3) Проведено попередню обробку та підготовку даних.
4) Побудовано модель машинного навчання та проведено її навчання на навчальному наборі даних.
5) Оцінено точність моделі на тестовому наборі даних, використовуючи метрики якості.
6) Виконано аналіз результатів, отриманих стосовно вхідного датасету. Вибрано алгоритм машинного навчання, який найкраще підходить для даної задачі.
7) Проведено моделювання застосунку для аналізу цін ноутбуків.
8) Розроблено застосунок для аналізу цін ноутбуків.
9) Проведено тестування розробленого застосунку та оцінено його ефективність та зручність для користувачів.
10) Оформлено звіту.

Було наведено сценарії використання вебзастосунку, а також створено діаграму активності та пакетів для наочної роботи застосунку та його функцій.

Було проведено докладний аналіз та підготовку набору даних для подальшого використання в моделі машинного навчання. Відзначено основні характеристики набору даних, виявлено взаємозв'язки та аномалії, та виконано передобробку даних,

зокрема обробку відсутніх значень, кодування категоріальних ознак, та масштабування числових ознак. Виявлено ключові аспекти, які будуть важливі при подальшому навчанні моделі.

У наступному етапі проведено навчання моделі та прогнозування цін.

Також представлено інтерфейс системи, який складається із двох блоків: візуалізація даних та прогнозування цін ноутбуків. Користувач має змогу обрати метод, а також метод візуалізації або модель для прогнозування ціни.

Отримані результати становлять основу для подальшого розвитку та навчання моделі машинного навчання з метою покращення прогнозування цін на ноутбуки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Laptops | Kaggle. URL: https://www.kaggle.com/datasets/anas123siddiqui/laptops (accessed 15/10/2023).

2. Machine Learning Algorithms For Beginners with Code Examples in Python | by Towards AI Editorial Team | Towards AI. URL: https://pub.towardsai.net/machine-learning-algorithms-for-beginners-with-python-code-examples-ml-19c6afd60daa (accessed 16/11/2023).

3. Random Forest Algorithm. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm (accessed 18/11/2023).

4. seaborn.boxplot — seaborn 0.12.2 documentation. URL: https://seaborn.pydata.org/generated/seaborn.boxplot.html (accessed 17/10/2023).

5. Top 10 Machine Learning Algorithms for Beginners | Built In. URL: https://builtin.com/data-science/tour-top-10-algorithms-machine-learning-newbies (accessed 16/11/2023).

6. Understanding the AdaBoost Algorithm | Built In. URL: https://builtin.com/machine-learning/adaboost (accessed 20/11/2023).

7. What is a use case diagram? URL: https://www.techtarget.com/whatis/definition/use-case-diagram (accessed 25/12/2024).

8. Your First Machine Learning Project in Python Step-By-Step - MachineLearningMastery.com. URL: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/ (accessed 17/11/2023).

9. O. Theobald. Machine Learning for Absolute Beginners: A Plain English Introduction Published, 2021. 180 pp..

10. Fit a Classification or Regression Tree : вебсайт. URL:

https://www.rdocumentation.org/packages/tree/versions/1.0-43/topics/tree (дата звернення: 20.11.2023).

11. Classification and Regression with Random Forest : вебсайт. URL: https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.1/topics/randomForest (дата звернення: 20.11.2023).

12. Generalized Boosted Regression Modeling (GBM) : вебсайт. URL: https://www.rdocumentation.org/packages/gbm/versions/2.1.8.1/topics/gbm (дата звернення: 20.11.2023).

13. Mean Squared Error : вебсайт. URL: https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_528 (дата звернення: 20.01.2024).

14. Mean Absolute Error : вебсайт. URL: https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_525 (дата звернення: 20.01.2024).

15. Root-mean-square deviation : вебсайт. URL: https://en.wikipedia.org/wiki/Root-mean-square_deviation (дата звернення: 20.01.2024).

16. Scikit-Learn: вебсайт. URL: https://scikit-learn.org/stable/ (дата звернення: 22.10.2023).

17. Pandas: вебсайт. URL: https://pandas.pydata.org/ (дата звернення: 22.10.2023).

18. NumPy: вебсайт. URL: https://numpy.org/ (дата звернення: 22.10.2023).

19. Matplotlib: вебсайт. URL: https://matplotlib.org/ (дата звернення: 22.10.2023).

ДОДАТОК А

VISUALIZATION

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-
theme.min.css">
    <link
href="https://fonts.googleapis.com/css?family=Lobster&subset=latin,cyri
llic" rel="stylesheet" type="text/css">

    <link rel="stylesheet" href="{% static 'css/My.css' %}">

    <title>User Interface</title>
</head>
<body>
    <header></header>

    <section>
        <form class="container" method="post" action="{% url
'interface_view' %}">
            <h1>User Interface</h1>
            {% csrf_token %}
            <label>Choose method:</label>
            <select name="method_choice" id="method_choice"
class="classic">
```

```
            <option value="visualization">Visualization</option>
            <option value="prediction">Prediction</option>
        </select>

        <div id="visualization_choices">
            <label>Choose visualization method:</label>
            <select name="visualization_choice"
id="visualization_choice" class="classic">
                <option value="plot_laptop_brands">Number of
laptops from different brands</option>
                <option value="plot_operating_systems">Operating
Systems</option>
                <option value="plot_ram_sizes">RAM Sizes</option>
                <option value="plot_average_cost_by_brand">Average
cost of laptops based on Brands</option>
                <option value="plot_average_cost_by_os">Average
cost of laptops based on Operating Systems</option>
                <option
value="plot_average_cost_by_processor_brands">Average cost of laptops
based on Processor Brands
                </option>
                <option
value="plot_average_cost_by_ram_size">Average cost of laptops based on
RAM</option>
                <option value="plot_share_of_laptops_by_os">Number
of laptops from different OS</option>
                <option
value="plot_share_of_laptops_by_processors">Number of laptops from
different Processors
                </option>
                <option
value="create_box_plot_min_max_price">Minimum and Maximum price for the
```

```
laptops based on the brand
                    </option>
                    <option
value="scatter_plot_os_processor">Combination of Laptop Brands and
Operating Systems available
                    </option>
                    <option value="heatmap_null_values">Null Values
Heatmap</option>
                    <option
value="distribution_plot_ram_size">Distribution of RAM Size</option>
                    <option
value="distribution_plot_of_ram">Distribution of Ram</option>
                    <option
value="distribution_plot_price">Distribution of price</option>
                    <option
value="bar_plot_median_price_by_brand">Median of Price by
brand</option>
                    <option
value="create_boxplot_ram_size_price">Boxplot of RAM size and
price</option>
                    <option
value="create_heatmap_correlation_price">Features Correlating with
Sales Price</option>
                </select>
            </div>

            <div id="prediction_choices" style="display: none">
                <label>Choose prediction method:</label>
                <select name="prediction_choice" id="prediction_choice"
class="classic">
                    <option value="linear_regression">Linear
Regression</option>
```

```
                    <option id="ridge" value="ridge">Ridge</option>
                    <option value="lasso">Lasso</option>
                    <option value="decision_tree_regressor">Decision
Tree Regressor</option>
                    <option value="sgd_regressor">SGD
Regressor</option>
                    <option value="k_neighbors_regressor">KNN
Regressor</option>
                    <option value="random_forest_regressor">Random
Forest Regressor</option>
                    <option value="ada_boost_regressor">AdaBoost
Regressor</option>
                    <option
value="gradient_boosting_regressor">Gradient Boosting
Regressor</option>
                    <option value="svr">SVR</option>
                    <option value="xgb_regressor">XGB
Regressor</option>
                    <option value="mlp_regressor">MLP
Regressor</option>
                    <option value="random_forest">Random
Forest</option>
                    <option value="random_forest">Random
Forest</option>
                    <option value="random_forest">Random
Forest</option>
                </select>
            </div>

            <button type="submit">Submit</button>
        </form>
```

```
        <div class="container">
            {% if method %}
                <h2>Result for {{ method }}:</h2>
                {% if result %}
                    {% if method|lower in visualization_methods %}
                        <img src="{{image_url }}" alt="{{ method }}">
                    {% else %}
                        <p>RESULT {{ result }}</p>
                        <p>Prediction result: {{
result.Prediction_result }}</p>
                        <p>MAE: {{result.MAE }}</p>
                        <p> R2 Score: {{ result.R2_Score}}</p>
                        <p>Execution Time: {{ result.Execution_time }}
seconds</p>
                        <p>PRED: {{ result.pred }}</p>
                        <p  id="Cross_Validation_Scores"
style="display: none;">Cross_Validation_Scores {{
result.Cross_Validation_Scores }}</p>
                        <p>Best_Ridge_Estimator {{
result.Best_Ridge_Estimator }}</p>
                        <p>Test_Score {{ result.Test_Score }}</p>
                        <p>Estimator {{ result.Estimator }}</p>
                    {% endif %}
                {% endif %}
            {% endif %}
        </div>
    </section>

    <script>
        // Show/hide choices based on the selected method

document.getElementById('method_choice').addEventListener('change',
```

```
function () {let visualizationChoices =
document.getElementById('visualization_choices');
            let predictionChoices =
document.getElementById('prediction_choices');
            let CrossValidationScores =
document.getElementById('Cross_Validation_Scores');

            if (this.value === 'visualization') {
                visualizationChoices.style.display = 'block';
                predictionChoices.style.display = 'none';
                CrossValidationScores.style.display = 'none';
            } else if (this.value === 'prediction') {
                visualizationChoices.style.display = 'none';
                predictionChoices.style.display = 'block';
                let predictionMethod =
document.getElementById('prediction_choices').value;
                if (predictionMethod === 'ridge') {
                    CrossValidationScores.style.display = 'block';
                } else {
                    CrossValidationScores.style.display = 'none';
                }
            } else {
                visualizationChoices.style.display = 'none';
                predictionChoices.style.display = 'block';
                CrossValidationScores.style.display = 'none';
            }
        });
    </script>

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