МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

"До захисту допущена"
Зав. кафедри комп'ютерних наук та прикладної математики

_____
«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

Застосування моделі машинного навчання для прогнозування популярності товарів на основі даних продажів з використанням бібліотеки scikit-learn

Виконав Мастило Роман
(прізвище, ім'я, по батькові) (підпис)
група ІПЗ – 41

Керівник: к. ек. н., доцент кафедри комп'ютерних наук та прикладної математики, Бачишина Л.Д.
(науковий ступінь, вчене звання, посада, прізвище, ініціали) (підпис)

Рівне – 2024

Національний університет водного господарства та природокористування

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

Кафедра **комп'ютерних наук та прикладної математики**

Освітньо-кваліфікаційний рівень **бакалавр**

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____

“____” _____20___року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

_____

1. Тема роботи *Застосування моделі машинного навчання для прогнозування популярності товарів на основі даних продажів з використанням бібліотеки scikit-learn.*

керівник роботи _____

затверджені наказом вищого навчального закладу від

2. Термін подання роботи студентом

3. Вихідні дані до роботи інтернет форуму та засоби розробки веб ресурсів.

4. Зміст розрахунково-пояснювальної записки:

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень) *мультимедійна презентація.*

6.Консультанти розділів роботи(проекту)

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділ 1	доцент Бачишина Л. Д.		
Розділ 2	доцент Бачишина Л. Д.		
Розділ 3	доцент Бачишина Л. Д.		

7. Дата видачі завдання : 08.01.24 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури за обраною тематикою	08.02. – 23.02.	
2	Опанування бібліотеки scikit-learn	07.03. – 20.03.	
3	Пошук набору даних	21.03 – 29.03.	
4	Тестування моделей	01.04. – 24.04.	
5	Створення інтерфейсу	05.05. – 31.05.	
6	Тестування програми	02.06. – 07.06.	
7	Підготовка виступу і презентації	15.06. – 19.06.	

Студент _____ _____

(підпис) (прізвище та ініціали)

Керівник роботи _____ _____

(підпис) (прізвище та ініціали)

ЗМІСТ

РЕФЕРАТ

Кваліфікаційна робота: с., малюнків, джерел

Метою кваліфікаційної роботи є розробка програми, яка за допомогою моделі машинного навчання прогнозує ціну на товари на основі даних з попередніх продажів

Об'єкт дослідження – прогнозування популярності товарів на основі історичних продажів, аналіз залежності між різними характеристиками товару та його продажами.

Предмет дослідження – методи машинного навчання для створення моделі прогнозування.

Методи дослідження – бібліотека skikit-learn, платформа PyCharm, python, flet.

Ключові слова:.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

МН(ML) –

ШІ(AI) –

NLP –

–

–

–

–

ВСТУП

В умовах сучасної економіки ефективне управління бізнесом значною мірою залежить від здатності компаній аналізувати великі обсяги даних і на основі цього прогнозувати майбутні тенденції. Зокрема, у роздрібній торгівлі одним із найважливіших завдань є прогнозування популярності товарів. Ця задача дозволяє оптимізувати запаси, зменшити витрати та збільшити прибутковість. Одним із найбільш ефективних підходів до вирішення такої задачі є використання методів машинного навчання (МН).

Машинне навчання, як напрямок штучного інтелекту, надає інструменти для автоматизації процесу аналізу даних та побудови прогнозних моделей. Завдяки здатності алгоритмів машинного навчання виявляти приховані закономірності в даних, можна створювати моделі, що дозволяють з високою точністю передбачати майбутні продажі товарів. Це, у свою чергу, дає змогу підприємствам ефективніше управляти асортиментом, планувати маркетингові кампанії та покращувати обслуговування клієнтів.

У рамках даної кваліфікаційної роботи розглядається застосування моделі машинного навчання для прогнозування популярності товарів на основі даних продажів, використовуючи бібліотеку scikit-learn. Scikit-learn – це одна з найпопулярніших бібліотек для машинного навчання в Python, яка надає широкий спектр інструментів для аналізу даних та побудови прогнозних моделей. Завдяки своїй простоті у використанні та гнучкості, scikit-learn дозволяє швидко і ефективно розробляти та тестувати різні моделі машинного навчання.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО МАШИННЕ НАВЧАННЯ,

1.1. Теоретичні аспекти машиного навчання

Машинне навчання (МН) – це підгалузь штучного інтелекту (ШІ), яка фокусується на розробці алгоритмів і статистичних моделей, що дозволяють комп'ютерам вчитися на даних без явного програмування. Ці алгоритми використовуються для виявлення шаблонів, прогнозування та прийняття рішень на основі даних.

МН як наукова галузь виросло з досліджень ШІ. На початку розвитку ШІ, деякі дослідники зосередилися на тому, щоб навчити машини обробляти дані. Вони використовували різні символьні методи та нейронні мережі, зокрема перцептрони, які згодом виявилися перевинаходами узагальнених лінійних моделей статистики. Також застосовувались ймовірнісні методи, особливо в медичній діагностиці.

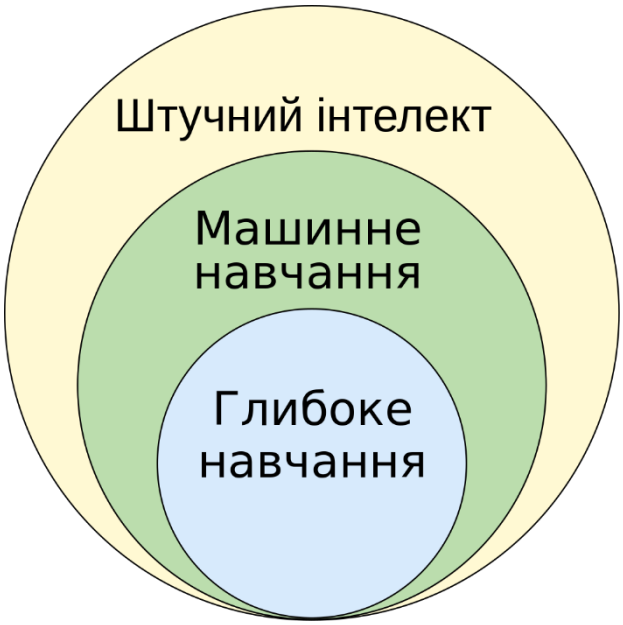

Рис.1.1 Машинне навчання як підгалузь.

Однак з часом основний акцент у ШІ змістився на логічні підходи, що призвело до розриву між ШІ та машинним навчанням. Імовірнісні методи мали

проблеми зі збиранням та поданням даних. Близько 1980 року домінуючими в ШІ стали експертні системи, а статистичні підходи втратили популярність. Символьні та знаннєві методи все ще використовувалися в ШІ, що призвело до індуктивного логічного програмування, але статистичні дослідження перейшли в області розпізнавання образів та інформаційного пошуку. Дослідження нейронних мереж також було залишено в рамках ШІ, проте продовжувалося в інших дисциплінах, зокрема у конекціонізмі, завдяки дослідникам, таким як Гопфілд, Румельхарт та Гінтон. В середині 1980-х років їхній головний успіх був пов'язаний із повторним відкриттям методу зворотного поширення.

У 1990-х роках машинне навчання, реорганізоване та визнане як окрема галузь, почало активно розвиватися. Основною метою стало вирішення практичних задач, а не досягнення штучного інтелекту. Фокус змістився від символьних методів до підходів, заснованих на статистиці, нечіткій логіці та теорії ймовірностей.

Машинне навчання знайшло застосування в багатьох сферах, таких як великі мовні моделі, комп'ютерне бачення, розпізнавання мовлення, фільтрація електронної пошти, сільське господарство та медицина. У цих галузях розробка алгоритмів для виконання певних завдань була б надто дорогою. У комерційних завданнях машинне навчання відоме як "передбачувальна аналітика". Хоча не все машинне навчання базується на статистиці, обчислювальна статистика є важливим джерелом методів для цієї галузі.

Математичні основи машинного навчання забезпечуються методами математичної оптимізації. Добування даних – це пов'язана сфера досліджень, яка зосереджена на аналізі даних через неконтрольоване навчання. З теоретичної точки зору, машинне навчання описується системою ймовірно приблизно коректного навчання.

1.2 Типи і основні алгоритми машинного навчання

Машинне навчання можна поділити на два основних типи: навчання на основі прецедентів (індуктивне навчання) та дедуктивне навчання. Оскільки дедуктивне навчання зазвичай відносять до сфери експертних систем, терміни «машинне навчання» та «навчання на основі прецедентів» часто використовують як синоніми. Сьогодні індуктивне навчання є дуже популярним, тоді як експертні системи стикаються з певними труднощами. Бази знань, на яких вони базуються, важко узгодити з реляційною моделлю даних, що ускладнює ефективне використання промислових СУБД для наповнення цих баз знань.

Навчання на основі прецедентів поділяють на три основні види: контрольоване навчання (навчання з учителем), неконтрольоване навчання (навчання без учителя) та навчання з підкріпленням.

Контрольоване навчання (supervised learning) передбачає використання набору прикладів з відомими відповідями для тренування моделі, яка потім може передбачати відповіді на нові приклади. Неконтрольоване навчання (unsupervised learning) працює з даними без попередньо відомих відповідей, шукаючи приховані закономірності та структури. Навчання з підкріпленням (reinforcement learning) включає взаємодію моделі з середовищем, отримуючи за це винагороди або покарання, що допомагає моделі вивчати оптимальну стратегію дій.

Окрім цих основних видів, розробляються й інші методи навчання, такі як активне навчання, багатозадачне навчання, різноманітне навчання та трансферне навчання. В останні роки особливо успішно розвивається глибоке навчання, яке може поєднувати алгоритми як з учителем, так і без учителя для досягнення високих результатів.

Контрольоване навчання . Це метод, який застосовується при роботі з великими обсягами даних, наприклад, з тисячами фотографій домашніх тварин, де кожна фотографія має маркер: кішка або собака. Завдання полягає в тому, щоб створити алгоритм, який зможе визначити, хто зображений на фотографії, яку він раніше не бачив – кішка або собака. Людина виступає в ролі "вчителя", заздалегідь позначаючи фотографії. Машина сама вибирає ознаки, за якими відрізняє кішок від собак. Знайдений алгоритм можна швидко адаптувати для вирішення інших завдань, наприклад, для розпізнавання курей і качок. Машина знову самостійно визначить ознаки, за якими розрізнятиме цих птахів. Нейромережу, навчена розпізнавати кішок, можна швидко переналаштувати на обробку результатів комп'ютерної томографії.

Алгоритми МН, що відповідають цьому навчанню:

- **Лінійна регресія** Використовується для прогнозування значення залежної змінної на основі значень незалежних змінних. Лінійна регресія намагається знайти найкращу пряму лінію, яка описує зв'язок між змінними.
- **Логістична регресія** Використовується для класифікації. Вона прогнозує ймовірність приналежності зразка до одного з класів (наприклад, ймовірність, що зображення є зображенням кішки).
- **Дерева рішень** Алгоритм, що розділяє дані на підгрупи на основі значень окремих ознак, створюючи дерево рішень. Використовується для класифікації та регресії.
- **Службовий вектор машин (SVM)** Алгоритм, який знаходить гіперплощину, що найкраще розділяє дані на класи.
- **k-ближчих сусідів (k-NN)** Алгоритм класифікації, який відносить зразок до того класу, до якого належать його найближчі сусіди.
- **Наївний баєсівський класифікатор** Алгоритм класифікації, заснований на теоремі Байеса, що передбачає незалежність ознак.

Неконтрольоване навчання. Хоча накопичено багато маркованих даних, даних без міток все ж більше. Це можуть бути зображення без підписів, аудіозаписи без коментарів або тексти без анотацій. Мета машини при неконтрольованому навчанні – виявити зв'язки між окремими даними, знайти закономірності, створити шаблони, впорядкувати дані або описати їх структуру, виконати класифікацію. Цей метод використовується, наприклад, у рекомендаційних системах, коли інтернет-магазин на основі аналізу попередніх покупок пропонує товари, які можуть зацікавити покупця більше за інші. Або коли після перегляду відео на YouTube користувачеві пропонують схожі відео. Або коли Google ранжує результати пошуку для різних користувачів по-різному, враховуючи їх історію пошуків.

Алгоритми МН, що відповідають цьому навчанню:

- **k-середніх (k-means)** Алгоритм кластеризації, який розбиває дані на k кластерів на основі схожості ознак.
- **Ієрархічна кластеризація** Алгоритм, який створює ієрархію кластерів, об'єднуючи найближчі кластери на кожному етапі.
- **Алгоритм головних компонент (PCA)** Алгоритм зниження розмірності, який знаходить нові змінні (головні компоненти), що зберігають максимальну дисперсію даних.
- **Алгоритм незалежних компонент (ICA)** Алгоритм зниження розмірності, який знаходить незалежні компоненти в даних.

Навчання з підкріпленням. Цей метод є окремим випадком контрольованого навчання, де "вчителем" виступає середовище. Машина, або "агент", не має попередньої інформації про середовище, але може виконувати в ньому різні дії. Середовище реагує на ці дії, надаючи агенту дані, які дозволяють йому

навчатися і реагувати на них. Таким чином, агент і середовище утворюють систему зі зворотним зв'язком.

Навчання з підкріпленням застосовується для вирішення більш складних завдань, ніж навчання з вчителем чи без нього. Наприклад, в системах навігації для роботів, які вчаться уникати зіткнень з перешкодами шляхом отримання зворотного зв'язку при кожному зіткненні. Також цей метод використовується в логістиці, при складанні графіків і плануванні завдань, а також у навчанні машин грати в логічні ігри, такі як покер, нарди, го та інші.

Алгоритми МН, що відповідають цьому навчанню:

- **Q-навчання**Алгоритм, що використовує таблицю значень (Q-таблицю) для зберігання найкращих дій в кожному стані, на основі винагороди.
- **Політичний градієнт (Policy Gradient)** Алгоритм, який безпосередньо оптимізує політику вибору дій, максимізуючи очікувану суму винагород.
- **Deep Q-Learning (глибоке Q-навчання)** Поєднання Q-навчання з глибокими нейронними мережами для вирішення складних завдань з великим числом станів і дій.

Нейронні мережі. Для машинного навчання використовуються різні технології та алгоритми, такі як дискримінантний аналіз і байєсовські класифікатори. Проте в кінці XX століття зросла увага до штучних нейронних мереж (ANN). Новий сплеск інтересу до них розпочався в 1986 році завдяки розвитку методу зворотного поширення помилки, який став успішним для навчання нейронних мереж.

Штучні нейронні мережі складаються зі з'єднаних і взаємодіючих штучних нейронів, створених на базі простих процесорів. Кожен процесор у мережі періодично отримує сигнали від інших процесорів, сенсорів або інших джерел

сигналів, і передає їх іншим процесорам. Ці прості процесори, об'єднані в мережу, здатні вирішувати складні завдання.

Зазвичай нейрони розташовані в мережі за рівнями або шарами. Нейрони першого рівня, як правило, є вхідними і отримують дані ззовні (наприклад, від сенсорів системи розпізнавання облич). Після обробки вони передають сигнали через синапси нейронам на наступному рівні. Нейрони другого рівня, який називають прихованим, обробляють ці сигнали і передають їх нейронам вихідного рівня. Оскільки кожен процесор вхідного рівня пов'язаний з кількома процесорами прихованого рівня, а ті, в свою чергу, пов'язані з процесорами вихідного рівня, така архітектура дозволяє ANN навчатися і знаходити прості взаємозв'язки в даних.

Глибоке навчання застосовується до більш складних ANN, які містять кілька прихованих рівнів. Рівні нейронів можуть чергуватися з шарами, що виконують складні логічні перетворення. Кожен наступний рівень мережі шукає взаємозв'язки в попередньому, що дозволяє мережі знаходити не тільки прості взаємозв'язки, але й зв'язки між цими взаємозв'язками.

Завдяки переходу до нейронних мереж з глибоким навчанням, компанія Google змогла значно підвищити якість свого продукту "Перекладач". Якість перекладу між англійською та французькою мовами підвищилася відразу на 7 балів, що на понад 20% більше. Попередня система, яка використовувала фразовий статистичний машинний переклад, досягла подібного поліпшення з 2006 року.

1.3 Застосування машинного навчання

Машинне навчання (МН) знаходить широке застосування в різних галузях, включаючи фінанси, медицину, маркетинг, транспорт і розваги.

У фінансах МН використовується для автоматизованих торгових систем, які аналізують ринки та здійснюють операції в реальному часі, забезпечуючи високу швидкість та точність. Також машинне навчання допомагає виявляти шахрайські дії шляхом аналізу фінансових транзакцій і виявлення аномалій, що знижує ризики і втрати.

У медицині МН застосовується для діагностики захворювань. Алгоритми аналізують медичні зображення, такі як рентгенівські знімки або МРТ, допомагаючи лікарям у виявленні патологій з високою точністю. Крім того, МН дозволяє прогнозувати результати лікування та розробляти персоналізовані плани лікування для пацієнтів на основі їхніх індивідуальних даних.

У сфері маркетингу машинне навчання використовується для створення рекомендаційних систем, які пропонують користувачам продукти та послуги на основі їхньої попередньої поведінки. Це дозволяє компаніям краще розуміти потреби клієнтів і підвищувати рівень задоволеності. Аналіз споживчих настроїв та сегментація ринку за допомогою МН також допомагає маркетологам розробляти більш ефективні стратегії просування товарів і послуг.

У транспорті МН використовується для розвитку автономних транспортних засобів, які здатні розпізнавати об'єкти на дорозі та приймати рішення в реальному часі, що забезпечує безпеку та ефективність руху. Машинне навчання також допомагає оптимізувати маршрути та управляти трафіком, що зменшує затори і підвищує ефективність транспортної системи.

У сфері розваг МН застосовується на стримінгових платформах для рекомендацій фільмів та музики. Алгоритми аналізують уподобання користувачів та пропонують контент, який може їх зацікавити, забезпечуючи більш персоналізований досвід.

Загалом МН можемо поділити на такі функції:

- Розпізнавання образів
- Обробка природної мови
- Рекомендаційні системи
- Автоматизовані ТЗ

Розпізнавання образів

Розпізнавання образів — це один із важливих напрямків машинного навчання, який дозволяє комп'ютерам ідентифікувати та класифікувати об'єкти візуальної інформації, такі як зображення або відео. Ця технологія знайшла широке застосування у багатьох сферах.

Наприклад, у медицині розпізнавання образів використовується для аналізу медичних зображень, допомагаючи лікарям виявляти захворювання, такі як рак або аномалії у внутрішніх органах. У сфері безпеки розпізнавання облич застосовується для ідентифікації людей, що дозволяє підвищити рівень безпеки в громадських місцях та на підприємствах.

У транспорті розпізнавання образів використовується в системах автономного водіння, де алгоритми аналізують дорожню обстановку, розпізнають знаки та перешкоди, що дозволяє автівкам рухатися без втручання водія. Також ця технологія допомагає в управлінні трафіком та контролі за дотриманням правил дорожнього руху.

В індустрії розваг розпізнавання образів застосовується для покращення досвіду користувачів, наприклад, в системах доповненої реальності, де користувачі можуть взаємодіяти з віртуальними об'єктами, інтегрованими в реальний світ.

Обробка природної мови

Обробка природної мови (NLP) — це галузь машинного навчання, яка займається взаємодією між комп'ютерами та людською мовою. Вона включає в себе методи та алгоритми, які дозволяють машинам розуміти, інтерпретувати та генерувати людську мову. NLP має безліч застосувань, які роблять її важливою для сучасних технологій.

Одним з основних напрямків NLP є розпізнавання мови, що дозволяє комп'ютерам конвертувати мовлення в текст. Це використовується в голосових помічниках, таких як Siri, Google Assistant або Alexa, де користувачі можуть взаємодіяти з пристроями за допомогою голосових команд. Системи розпізнавання мови застосовуються також у телефонних центрах для автоматичного обслуговування клієнтів.

Іншим важливим напрямком є аналіз тексту. NLP дозволяє комп'ютерам розуміти зміст тексту, аналізувати його структуру та визначати емоційне забарвлення. Це застосовується в системах аналізу відгуків клієнтів, де компанії можуть автоматично визначати настрої користувачів щодо їх продуктів чи послуг. Аналіз тексту також використовується в юридичній сфері для автоматичного аналізу документів та виявлення важливої інформації.

Машинний переклад — ще один важливий аспект NLP. Завдяки таким системам, як Google Translate, комп'ютери можуть перекладати текст з однієї мови на іншу з досить високою точністю. Це полегшує комунікацію між людьми, які говорять різними мовами, і робить інформацію доступнішою на глобальному рівні.

Синтез мови дозволяє комп'ютерам генерувати мовлення на основі тексту. Ця технологія застосовується у навігаційних системах, де пристрої озвучують інструкції для водіїв, а також у програмах для людей з порушеннями зору, які озвучують текстові повідомлення або веб-сторінки.

NLP також використовується для автоматичного реферування текстів, де алгоритми можуть виділяти основні тези та створювати короткі резюме довгих документів. Це корисно в інформаційних агентствах, де потрібно швидко обробляти великі обсяги інформації, або в академічних дослідженнях для аналізу наукових статей.

Чат-боти **і** віртуальні асистенти є ще одним популярним застосуванням NLP. Вони використовують обробку природної мови для взаємодії з користувачами в режимі реального часу, відповідаючи на запитання, надаючи підтримку або виконуючи завдання. Це значно підвищує ефективність обслуговування клієнтів та забезпечує доступ до інформації 24/7.

Рекомендаційні системи

Рекомендаційні системи є однією з найпотужніших та найвпливовіших технологій, що базуються на машинному навчанні та обробці даних. Вони створені для надання персоналізованих рекомендацій користувачам, враховуючи їхні вподобання, історію взаємодій та поведінку. Ці системи знайшли широке застосування в багатьох галузях, включаючи електронну комерцію, стрімінгові сервіси, соціальні мережі, онлайн-навчання та багато інших.

Рекомендаційні системи можна поділити на кілька типів, кожен з яких використовує різні методи для створення рекомендацій. Системи на основі фільтрації контенту аналізують властивості самих продуктів або контенту і рекомендують користувачам подібні товари або матеріали. Наприклад, якщо користувач прочитав статтю про здоровий спосіб життя, система може запропонувати йому інші статті на цю ж тему.

Системи на основі колаборативної фільтрації базуються на аналізі взаємодій користувачів з продуктами або контентом. Такий підхід дозволяє надавати рекомендації на основі того, що інші користувачі зі схожими вподобаннями

також переглянули та високо оцінили певний фільм чи продукт. Це дозволяє враховувати не лише інтереси окремого користувача, а й досвід всієї спільноти користувачів.

Гібридні системи поєднують елементи фільтрації контенту та колаборативної фільтрації, що забезпечує більш точні та релевантні рекомендації. Вони можуть враховувати як властивості продуктів, так і поведінку користувачів, що робить рекомендації більш персоналізованими та корисними.

Рекомендаційні системи широко використовуються в різних сферах. Наприклад, в електронній комерції вони допомагають користувачам знаходити продукти, які можуть їх зацікавити, що підвищує задоволеність клієнтів і збільшує продажі. Стрімінгові сервіси, такі як Netflix або Spotify, використовують рекомендаційні системи для підбору фільмів, серіалів або музики, що відповідають вподобанням користувачів, тим самим збільшуючи час, проведений на платформі, та покращуючи досвід користувачів.

У соціальних мережах рекомендаційні системи допомагають знаходити цікавий контент і нових друзів, а також підтримувати зв'язок із вже існуючими. В онлайн-навчанні такі системи використовуються для підбору навчальних матеріалів та курсів, що відповідають рівню знань і інтересам студентів, що робить навчання більш ефективним і приємним.

Загалом, рекомендаційні системи є невід'ємною частиною сучасних технологій, які значно покращують взаємодію користувачів з різними платформами та сервісами, роблячи їх більш персоналізованими та зручними.

Автоматизовані транспортні засоби

Автоматизовані транспортні засоби (АТЗ), або автономні автомобілі, є однією з найбільш перспективних і революційних технологій сучасності. Вони здатні самостійно пересуватися дорогами, використовуючи поєднання різноманітних

сенсорів, алгоритмів машинного навчання та складних систем управління. Розвиток АТЗ обіцяє значно змінити наше життя, підвищуючи безпеку дорожнього руху, зменшуючи затори і роблячи транспортні системи більш ефективними.

Проте, незважаючи на значний прогрес, автономні транспортні засоби все ще стикаються з численними викликами. Серед них — забезпечення безпеки та надійності систем, адаптація до різних погодних умов та дорожніх ситуацій, а також вирішення правових і етичних питань, пов'язаних з відповідальністю за аварії.

Перспективи розвитку АТЗ є дуже обнадійливими. З кожним роком технології стають все більш досконалими, і ми наближаємося до того дня, коли автономні автомобілі стануть звичайним явищем на дорогах. Вони обіцяють зробити наше життя безпечнішим, зручнішим і більш екологічним.

Машинне навчання для бізнесу

Ринок машинного навчання стрімко розвивається. З 2016 року його обсяг перевищив $1 мільярд, а до 2025 року, за прогнозами, він може зрости до $39,98 мільярда.

В кінці 2016 року MIT Technology Review та Google Cloud провели спільне дослідження під назвою "Машинне навчання: новий спосіб отримати конкурентну перевагу". В опитуванні взяли участь 375 кваліфікованих респондентів з різних країн світу, які працюють в малих та великих компаніях з різних галузей, таких як промисловість, послуги та фінанси. Результати дослідження показали, що 60% компаній вже використовують машинне навчання (ML), а третина з них перейшла від інноваційного етапу до стадії зрілості. Більше того, 26% компаній вже отримують конкурентну перевагу завдяки ML. Чверть компаній інвестують в ML понад 15% своїх IT-бюджетів і значною мірою повертають зроблені інвестиції.

Машинне навчання і нейронні мережі ефективні для вирішення бізнес-завдань у таких випадках:

- Коли накопичено велику кількість різноманітних даних, але відсутні програми для їх обробки та систематизації.
- Коли дані є спотвореними, неповними або несистематизованими.
- Коли дані настільки різнорідні, що важко виявити зв'язки і закономірності між ними.

Машинне навчання та нейронні мережі можуть вирішувати такі бізнес-завдання:

- *Прогнозування*: попиту, обсягу продажів, запасів на складі, завантаженості обладнання та інших ресурсів, подальшого розвитку підприємства.
- *Виявлення*: тенденцій, прихованих взаємозв'язків, аномалій, повторюваних елементів.
- *Розпізнавання*: фото-, відео-, аудіоконтенту, спроб шахрайства, брехні, внутрішніх загроз, зовнішніх атак на систему безпеки.
- *Автоматизація:* роботи операторів в онлайн-чатах, телефонних операторів.
- *Класифікація*: аналіз складу покупців, клієнтів, замовників та їх сегментація за різними параметрами.
- *Кластеризація*: класифікація за параметрами, які спочатку не були відомі.
- *Розробка*: чат-ботів для покращення взаємодії з клієнтами.

Таким чином, машинне навчання стає невід'ємною частиною сучасного бізнесу, допомагаючи компаніям ефективно використовувати дані, підвищувати продуктивність та отримувати конкурентні переваги.

Найбільша у світі торгова платформа Alibaba активно використовує машинне навчання та інші технології штучного інтелекту. Це дозволяє її віртуальним вітринам адаптуватися під кожного покупця, а система пошуку надає найбільш релевантні варіанти товарів. Крім того, чат-бот Ali Xiaomi здатний самостійно вирішувати більшість запитів клієнтів у техпідтримку. Розроблена Alibaba нейронна мережа вперше перевершила людські результати в тестах Стенфордського університету, які включають завдання на читання або прослуховування інформації з подальшими перевірочними питаннями.

Американська торгова мережа Target виявила, що за допомогою машинного навчання можна не лише прогнозувати поведінку покупців, але й визначати зміни в їхньому житті, такі як вагітність. Алгоритми Target настільки точні, що можуть визначити триместр вагітності жінки за її покупками.

Популярний фотохостинг Pinterest застосовує машинне навчання для підбору найцікавіших фотографій для своїх користувачів, що значно покращує їхній досвід користування сервісом.

Лукас Бівальд, генеральний директор компанії Figure Eight (колишня CrowdFlower), яка займається проєктами машинного навчання, зазначає, що ця технологія вже суттєво змінює роботу багатьох фірм. І над цим працюють не тільки великі компанії, як Google або Microsoft, але й всі компанії зі списку Fortune 500, які завдяки машинному навчанню працюють ефективніше та заробляють більше.

Серед компаній з українським корінням варто відзначити стартап Neuromation, який у лютому 2017 року під час ICO залучив $71,6 млн інвестицій. Платформа Neuromation дозволяє створювати штучні навчальні середовища для глибокого навчання нейронних мереж на великій кількості прикладів. Дані для навчання генеруються за допомогою обчислювальних потужностей блокчейн-спільноти. Це рішення виникло через нестачу обчислювальних ресурсів під

час роботи над системами комп'ютерного зору. Оренда ресурсів у хмарних сервісів Amazon або Google була надто дорогою для стартапу, а через бум майнінгу було важко придбати відеокарти. Так виникла ідея орендувати обчислювальні потужності у майнерів, що зрештою призвело до створення нейроплатформи.

РОЗДІЛ 2. БІБЛІОТЕКА SCIKIT-LEARN ТА ПОБУДОВА МОДЕЛІ

2.1. Бібліотека scikit-learn

Scikit-learn — це одна з найпопулярніших і найширше використовуваних бібліотек для машинного навчання на мові програмування Python. Вона пропонує інструменти для побудови та тренування моделей машинного навчання, аналізу даних та передбачення результатів. Scikit-learn призначена для розв'язання задач як контрольованого, так і неконтрольованого навчання і відрізняється своєю простотою у використанні, потужністю та ефективністю.

Scikit-learn має велику та активну спільноту користувачів і розробників, яка постійно сприяє її розвитку та вдосконаленню. Документація бібліотеки добре структурована і містить багато прикладів, що робить її легкою для вивчення і використання навіть для початківців.

Завдяки своїй простоті, потужності та гнучкості, Scikit-learn стала незамінним інструментом для науковців, інженерів та аналітиків, які працюють в галузі машинного навчання та аналізу даних.

Історія розвитку

Scikit-learn — це популярна бібліотека для машинного навчання, розроблена для мови програмування Python. Її розробка почалася як частина проекту Google Summer of Code в 2007 році. Проект був ініційований Давідом Куртоном (David Cournapeau), і перші версії бібліотеки були створені під

назвою "scikits.learn". Мета проекту полягала в тому, щоб створити бібліотеку, яка була б простою у використанні, але водночас потужною та ефективною.

У 2010 році бібліотека пройшла серйозну переробку за участі Матьє Блаше (Mathieu Blondel), Гільєма Лелега (Gaël Varoquaux) та інших розробників з INRIA (Французький національний інститут досліджень у сфері комп'ютерних наук та автоматики). Вони допомогли стандартизувати API бібліотеки та додати багато нових функцій. З того часу Scikit-learn швидко зростала, і зараз вона є однією з найпопулярніших бібліотек для машинного навчання у світі.

Переваги Scikit-learn

1. Простота використання: Scikit-learn має інтуїтивно зрозумілий API, який дозволяє легко створювати та тренувати моделі машинного навчання. Це робить її доступною для новачків і зручною для досвідчених розробників.
2. Комплексний набір інструментів: Бібліотека включає широкий спектр алгоритмів машинного навчання для класифікації, регресії, кластеризації, зниження розмірності та ансамблевих методів. Це дозволяє вирішувати різні задачі з використанням одного інструменту.
3. Інтеграція з іншими бібліотеками: Scikit-learn побудована на основі таких популярних бібліотек, як NumPy, SciPy та Matplotlib, що забезпечує її інтеграцію з іншими інструментами для обробки даних і візуалізації.
4. Відмінна документація: Документація Scikit-learn є дуже добре структурованою і містить багато прикладів та посібників. Це допомагає користувачам швидко освоїти бібліотеку та знайти відповіді на свої питання.
5. Активна спільнота: Scikit-learn має велику та активну спільноту користувачів та розробників, що сприяє постійному розвитку та

вдосконаленню бібліотеки. Користувачі можуть отримати підтримку та обмінюватися знаннями через різні форуми та групи.

Недоліки Scikit-learn

1. Обмеження щодо великих даних: Scikit-learn найкраще підходить для роботи з невеликими та середніми наборами даних. Для обробки дуже великих обсягів даних можуть знадобитися спеціалізовані інструменти, такі як Apache Spark або Dask.
2. Відсутність підтримки глибокого навчання: Scikit-learn не призначена для побудови та тренування глибоких нейронних мереж. Для таких задач краще використовувати бібліотеки, такі як TensorFlow або PyTorch.
3. Відносно повільна швидкість обчислень: У порівнянні з іншими бібліотеками для машинного навчання, деякі алгоритми Scikit-learn можуть працювати повільніше. Це може бути особливо помітно при роботі з великими наборами даних або складними моделями.
4. Обмежена підтримка онлайн-навчання: Scikit-learn більше орієнтована на пакетне навчання і має обмежену підтримку алгоритмів онлайн-навчання, які обробляють дані поступово, по мірі їх надходження.

Отже, Scikit-learn є потужною і зручною бібліотекою для машинного навчання на Python, яка підходить для широкого спектра задач. Її простота, багатий набір інструментів та активна спільнота роблять її відмінним вибором для розробників, дослідників і аналітиків. Незважаючи на деякі обмеження, Scikit-learn залишається одним з найпопулярніших інструментів для машинного навчання завдяки своїй ефективності та гнучкості.

2.2 Підготовка та обробка даних

Обробка даних є ключовим етапом у машинному навчанні та аналітиці даних, оскільки дозволяє створювати точні, ефективні та надійні моделі. Сирі дані часто містять помилки, пропуски, шум або непотрібну інформацію. Обробка даних включає очистку, видалення аномалій та заповнення пропущених значень, що підвищує якість даних і, відповідно, точність моделей. Крім того, високовимірні дані можуть бути складними для аналізу. Зменшення розмірності даних шляхом відбору ознак або застосування методів, таких як PCA (метод головних компонент), спрощує моделі та зменшує обчислювальні витрати.

Добре оброблені дані дозволяють моделям машинного навчання працювати ефективніше та швидше, оскільки вони позбавлені зайвої інформації та зосереджені на ключових аспектах. Це також допомагає уникнути переобучення моделей, що виникає, коли модель занадто добре адаптується до тренувальних даних і погано узагальнює нові дані.

Процес обробки даних у машинному навчанні починається зі збору сирих даних з різних джерел, таких як бази даних, файли CSV, API або веб-скрапінг. Після цього дані очищуються від помилок, заповнюються пропущені значення, обробляються аномалії та видаляються дублікати. Наступним кроком є попередня обробка даних, яка включає нормалізацію або стандартизацію числових даних, кодування категоріальних змінних (наприклад, one-hot encoding), а також розбиття датасету на тренувальну і тестову вибірки.

Інженерія ознак, яка полягає у створенні нових ознак з існуючих даних, є важливою складовою обробки даних. Це може включати обчислення відстаней, часу тощо. Відбір найбільш релевантних ознак допомагає зменшити розмірність даних та уникнути зайвої інформації. Розбиття даних на

тренувальні та тестові вибірки зазвичай здійснюється в пропорції 70/30 або 80/20. Трансформація даних за допомогою методів, таких як PCA або LDA, допомагає зменшити розмірність та спростити дані.

Обробка даних – це критично важливий етап, який визначає успіх всього проекту машинного навчання. Без належної обробки навіть найкращі алгоритми можуть працювати погано.

В обробці даних для машинного навчання існує кілька ключових етапів, кожен з яких важливий для успіху моделі. Основні види обробки даних:

1. Збір даних
2. Очищення даних
3. Перертворення даних
4. Генерація нових ознак
5. Розподіл даних
6. Обробка часових рядів
7. Обробка зображень
8. Обробка звукових даних

Збір даних

Збір даних є першим і одним з найважливіших етапів у процесі машинного навчання. Від якості та кількості зібраних даних залежить якість майбутньої моделі. Дані можна збирати з різних джерел. Серед них бази даних, такі як SQL і NoSQL (наприклад, MongoDB), що містять структуровані дані. Автоматичне збирання даних з веб-сайтів за допомогою інструментів, таких як BeautifulSoup або Scrapy, також є ефективним способом. Використання інтерфейсів програмування додатків (API), таких як Twitter API чи Google Maps API, дозволяє отримувати дані з веб-сервісів. Іншим способом збору даних є проведення опитувань або інтерв’ю, що дає змогу отримати дані

безпосередньо від людей. Аналіз лог-файлів веб-серверів або додатків дозволяє отримувати дані про поведінку користувачів.

Готові датасети можна знайти на відкритих ресурсах, таких як Kaggle, UCI Machine Learning Repository і Google Dataset Search. Зібрані дані можуть бути збережені на локальних комп'ютерах або серверах, а також у хмарних сховищах, таких як AWS, Google Cloud або Azure, що дозволяє зберігати великі обсяги даних і забезпечує їх доступність для обробки.

Очищення даних

Очищення даних є критично важливим етапом, який полягає у підготовці даних до аналізу шляхом видалення або виправлення помилок, пропусків та інших аномалій. Перш за все, необхідно видалити дублікати, оскільки їх наявність може спотворити результати аналізу. Потім варто звернути увагу на пропущені значення. Їх можна або видалити, якщо їх кількість значна, або заповнити середніми чи медіанними значеннями, або ж скористатися спеціалізованими алгоритмами для передбачення пропущених даних.

Також важливо виправити аномалії в даних. Викиди (аномальні значення) можуть серйозно впливати на точність моделі, тому їх необхідно виявити і, за можливості, виправити або видалити. Інколи доцільно застосувати методи логарифмування або нормалізації для зменшення впливу аномалій. Перетворення даних також відіграє важливу роль. Масштабування дозволяє привести всі дані до єдиного масштабу, що запобігає домінуванню однієї ознаки над іншими. Категоріальні дані часто потребують перетворення у числові формати, що можна здійснити за допомогою one-hot encoding чи label encoding.

Форматування даних включає приведення їх до необхідного вигляду, наприклад, форматування дат чи перетворення тексту до нижнього регістру.

Останнім кроком є виявлення та виправлення помилок, таких як орфографічні, логічні чи синтаксичні помилки, що забезпечує цілісність і точність даних.

Очищення даних є важливим етапом, оскільки якість підготовлених даних безпосередньо впливає на точність і надійність моделей машинного навчання.

Перетворення даних

Перетворення даних є критичним етапом, який допомагає зробити дані більш придатними для аналізу і покращити результати моделі машинного навчання. Масштабування даних – це одна з основних процедур, яка приводить всі ознаки до одного масштабу. Це необхідно для уникнення ситуацій, коли одна ознака домінує над іншими через свої великі числові значення. Нормалізація та стандартизація – два популярних методи масштабування.

Категоріальні ознаки потребують особливої уваги, оскільки їх необхідно перетворити в числовий формат. Один з методів – це one-hot encoding, коли кожне категоріальне значення перетворюється в окрему колонку з бінарними значеннями (0 або 1). Інший метод – label encoding, коли кожна категорія кодується унікальним числовим значенням. Також важливо розглянути логарифмування та інші нелінійні перетворення, які можуть допомогти зменшити асиметрію розподілу ознак та зробити їх більш нормальними.

Генерація нових ознак

Генерація нових ознак (Feature Engineering) є важливим етапом, який може значно покращити продуктивність моделі. Це включає створення нових ознак на основі існуючих. Наприклад, з дати можна виділити такі ознаки, як рік, місяць, день тижня, що може бути корисним для моделювання сезонних тенденцій.

Поліноміальні ознаки – це ще один підхід, який включає додавання квадратів, кубів та інших ступенів вихідних ознак. Це дозволяє моделі виявляти більш складні взаємозв'язки між ознаками. Також можна використовувати методи зменшення розмірності, такі як PCA (Principal Component Analysis) або LDA (Linear Discriminant Analysis), які допомагають зменшити кількість ознак, зберігаючи при цьому максимальну кількість інформації. Це може бути особливо корисним, коли дані мають велику кількість ознак, що може призвести до проблеми перенавчання.

Розподіл даних

Розподіл даних на тренувальні, валідаційні та тестові вибірки є важливим етапом у процесі машинного навчання, який допомагає оцінити продуктивність моделі та уникнути перенавчання. Тренувальна вибірка використовується для навчання моделі, тоді як валідаційна вибірка допомагає налаштувати гіперпараметри та вибрати найкращу модель. Тестова вибірка, яка не використовується під час навчання, дозволяє оцінити кінцеву продуктивність моделі на нових даних.

Крос-валідація є популярним методом для більш надійної оцінки якості моделі. Вона включає розподіл даних на кілька підгруп (folds) і повторне тренування моделі на кожній з них, при цьому кожна підгрупа по черзі використовується як тестова вибірка. Це дозволяє зменшити варіативність оцінок та отримати більш точну оцінку продуктивності моделі.

Розподіл даних допомагає забезпечити надійність і стабільність моделі, дозволяючи уникнути перенавчання та перевірити, як добре модель узагальнює знання на нових, невідомих даних.

Обробка часових рядів

Обробка часових рядів є спеціалізованим етапом, який включає роботу з даними, що мають часовий компонент. Часові ряди часто зустрічаються в фінансових, економічних, кліматичних та багатьох інших прикладних областях. Перш за все, важливо перетворити часові дані у підходящий формат, що дозволяє правильно працювати з ними в аналізі та моделях. Зазвичай, це включає перетворення текстових дат в формат дати-часу.

Виділення часових ознак є ще одним важливим кроком. З основних дат можна отримати такі ознаки, як рік, місяць, день, година, хвилина тощо. Крім того, часто використовуються додаткові ознаки, такі як ковзне середнє, що допомагає згладжувати короткострокові коливання та виділяти довгострокові тенденції. Сезонність та тренди також можуть бути враховані шляхом створення відповідних ознак, що відображають циклічні зміни в даних.

Обробка зображень

Обробка зображень є важливою частиною роботи з візуальними даними, особливо у комп’ютерному зорі та аналізі зображень. Перший крок – це зміна розмірів зображень для приведення їх до єдиного формату, що необхідно для конволюційних нейронних мереж. Нормалізація пікселів, яка включає перетворення значень пікселів в діапазон від 0 до 1, також є важливим етапом, оскільки допомагає зменшити вплив варіативності освітлення та контрасту.

Аугментація даних є методом штучного збільшення кількості зображень в тренувальному наборі. Це включає такі операції, як повороти, масштабування, обертання, зрізання та відображення зображень. Аугментація дозволяє моделі стати більш стійкою до різноманітних варіацій вхідних даних і покращити її здатність до узагальнення.

Обробка звукових даних

Обробка звукових даних включає кілька етапів, що дозволяють перетворити звукові сигнали у форму, придатну для аналізу. Перший етап – це виділення ознак. Однією з найпоширеніших ознак є мел-кепстральні коефіцієнти (MFCC), які представляють спектральну енергію в різних частотних діапазонах і часто використовуються у розпізнаванні мови та звуків.

Перетворення звукових сигналів у спектрограми є ще одним важливим кроком. Спектрограми дозволяють візуалізувати частотний вміст сигналу у часі, що допомагає моделі краще розпізнавати патерни та характеристики звуку. Такі методи, як короткочасне перетворення Фур'є (STFT), дозволяють отримувати спектрограми зі звукових сигналів. Це є основою для подальшого аналізу та використання у різноманітних додатках, таких як розпізнавання мови, музики або акустичних подій.

Обробка звукових даних, як і обробка зображень, вимагає спеціалізованих методів і інструментів, що дозволяють ефективно працювати з такими типами даних і отримувати якісні результати в моделях машинного навчання.

2.3. Методи оцінки моделей

Оцінка моделей машинного навчання є важливою для розуміння їхньої продуктивності та визначення того, наскільки добре модель працюватиме на нових даних. Ось основні методи та метрики, що використовуються для оцінки моделей машинного навчання:

Метод розділення даних на навчальну та тестову вибірки полягає у поділі наявного набору даних на дві частини: навчальну вибірку (training set) і тестову вибірку (test set). Зазвичай, співвідношення між ними становить 70:30, 80:20 або 90:10, залежно від розміру та специфіки даних. Спочатку дані очищуються та нормалізуються для підготовки до поділу. Потім дані розділяються на навчальну та тестову вибірки випадковим чином, щоб

забезпечити репрезентативність. Модель навчається на навчальній вибірці, а потім тестується на тестовій вибірці для оцінки її продуктивності. Цей метод простий у реалізації та швидкий у виконанні, але може призвести до зміщених оцінок, якщо вибірки не будуть репрезентативними.

Крос-валідація є більш надійним методом оцінки моделей, оскільки вона використовує всі доступні дані як для навчання, так і для тестування. Найпопулярнішим методом є k-кратна крос-валідація (k-fold cross-validation). Спочатку дані випадковим чином розбиваються на k рівних частин (фолдів). Модель навчається на (k-1) фолдах і тестується на одному фолді. Процес повторюється k разів, кожного разу з іншим тестовим фолдом. Результати оцінки з кожного фолду усереднюються для отримання остаточної оцінки моделі. Крос-валідація забезпечує більш надійну оцінку продуктивності моделі, оскільки всі дані використовуються як для навчання, так і для тестування. Вона також зменшує ризик отримання зміщених результатів через випадковий поділ даних. Однак, цей метод є більш витратним з точки зору обчислювальних ресурсів та часу порівняно з простим розділенням на навчальну та тестову вибірки і складніший у реалізації.

Існують також інші варіації крос-валідації, такі як стратифікована крос-валідація (stratified cross-validation), яка забезпечує, що кожен фолд має пропорційне представництво класів, що особливо важливо для незбалансованих даних.

Метрики точності є важливим аспектом оцінки моделей машинного навчання, оскільки вони дозволяють зрозуміти, наскільки добре модель виконує свою задачу. Розглянемо метрики точності для двох основних типів задач: класифікація та регресія.

Для класифікації метрики точності допомагають оцінити, наскільки правильно модель класифікує дані. Однією з найпоширеніших метрик є точність

(accuracy)**,** яка визначає частку правильно передбачених класів серед усіх передбачень. Однак, ця метрика може бути недостатньо інформативною для незбалансованих даних. У таких випадках використовуються метрики полнота (recall) **і** точність (precision)**.** Полнота вимірює частку правильних позитивних передбачень серед усіх істинних позитивних випадків, тоді як точність вимірює частку правильних позитивних передбачень серед усіх передбачених позитивних випадків. Ці метрики об'єднуються у F-міру (F1 score)**,** яка є гармонійним середнім між точністю і полнотою, що дозволяє отримати збалансовану оцінку продуктивності моделі.

Іншою важливою метрикою для класифікації ***є матриця помилок (confusion matrix)*****.** Вона показує кількість істинних позитивних, хибних позитивних, істинних негативних і хибних негативних передбачень. Ця матриця дозволяє детальніше аналізувати помилки моделі і визначати, які класи найчастіше плутаються.

Крива ROC (Receiver Operating Characteristic) і *AUC (Area Under the Curve)* також є важливими інструментами для оцінки моделей класифікації. Крива ROC показує співвідношення між полнотою і хибнопозитивними передбаченнями, тоді як AUC вимірює площу під кривою ROC. AUC є зручним для порівняння різних моделей, оскільки вона дає єдине числове значення, що відображає загальну продуктивність моделі.

Для регресії метрики точності оцінюють, наскільки добре модель прогнозує числові значення. Однією з основних метрик є *середньоквадратична помилка (Mean Squared Error, MSE)***,** яка обчислює середнє значення квадратів різниць між передбаченими і фактичними значеннями**.** *Середня абсолютна помилка (Mean Absolute Error, MAE)* вимірює середнє значення абсолютних різниць між передбаченими і фактичними значеннями, що дозволяє зрозуміти середню помилку прогнозів моделі. *Коефіцієнт детермінації* (R^2) показує, яка частка

дисперсії залежної змінної пояснюється незалежними змінними, і є показником, наскільки добре модель підходить до даних.

Ці метрики допомагають розробникам машинного навчання оцінювати продуктивність моделей та вибирати найбільш підходящі для конкретних задач.

Матриця помилок є інструментом для оцінки моделей класифікації, який дозволяє детально аналізувати їхні помилки. Вона представляє собою таблицю, де по осі Y знаходяться реальні класи, а по осі X — передбачені класи. Кожен елемент матриці показує кількість передбачень для конкретного поєднання реального та передбаченого класів.

Матриця помилок складається з чотирьох основних компонентів:

- Істинні позитивні (True Positives, TP) — кількість правильних передбачень позитивного класу.
- Хибні позитивні (False Positives, FP) — кількість неправильних передбачень позитивного класу.
- Істинні негативні (True Negatives, TN) — кількість правильних передбачень негативного класу.
- Хибні негативні (False Negatives, FN) — кількість неправильних передбачень негативного класу.

Ці компоненти дозволяють обчислити різні метрики точності, такі як точність, полнота, точність передбачень, F1-міра та інші. Матриця помилок надає зрозумілий спосіб оцінити, які саме помилки найчастіше допускає модель, і які класи найчастіше плутаються між собою.

Крива ROC (Receiver Operating Characteristic) — це графічне представлення продуктивності моделі класифікації, яке показує співвідношення між полнотою (чутливістю) і частотою хибнопозитивних

передбачень (1 - специфічність) при різних порогах класифікації. Крива ROC дозволяє оцінити, як добре модель відокремлює класи один від одного на всьому діапазоні можливих порогів.

Для побудови кривої ROC на осі X відкладається частота хибнопозитивних передбачень (False Positive Rate), а на осі Y — полнота (True Positive Rate). Ідеальна модель має криву, що проходить через верхній лівий кут (координати (0,1)), що означає максимальну полноту при мінімальній частоті хибнопозитивних передбачень.

Крива AUC (Area Under the Curve) — це площа під кривою ROC. Вона дає єдине числове значення для оцінки продуктивності моделі: чим ближче значення AUC до 1, тим краща модель. AUC = 0.5 означає, що модель не краще випадкового передбачення, тоді як AUC = 1.0 вказує на ідеальну модель.

Логарифмічна функція втрат, також відома як кросс-ентропійна втрата, використовується для оцінки моделей класифікації, які передбачають ймовірності приналежності до певного класу. Ця метрика вимірює невизначеність передбачених імовірностей відносно справжніх класів.

Чим нижче значення, тим краща модель. Ідеальна модель, яка точно передбачає ймовірності, матиме Log Loss, близьку до 0. Високе значення вказує на те, що модель погано передбачає ймовірності, тобто сильно відхиляється від справжніх класів.

Логарифмічна функція втрат є важливою, тому що вона карає великі помилки сильніше, ніж менші. Наприклад, якщо модель передбачає ймовірність 0.01 для позитивного класу, тоді як справжній клас — 1, це призведе до великого збільшення Log Loss.

РОЗДІЛ 3. СТВОРЕННЯ ТА ОГЛЯД ПРОГРАМИ

3.1. Пошук підходящого набору даних

Для початку мені навчання мені треба знайти підходящий набір даних. Я роглядав такі ресурси:

Kaggle. Ця платформа пропонує безліч відкритих датасетів різних категорій, включаючи дані про продажі, маркетинг та поведінку споживачів. Ви можете знайти датасети, які вже містять історичні дані про продажі різних товарів, що дозволить вам провести аналіз та створити модель для прогнозування популярності.

Ще одним корисним ресурсом є **UCI Machine Learning Repository**, де зберігаються різноманітні набори даних для машинного навчання. Цей репозиторій містить багато датасетів, які можуть бути корисними для аналізу продажів і прогнозування попиту.

Також звернув увагу на **Google Dataset Search**, інструмент, розроблений для пошуку датасетів по всьому інтернету. З його допомогою можна знайти різні набори даних, пов'язані з продажами, ринковими тенденціями та споживчою поведінкою.

Крім того, розглядав використання відкритих даних з **Amazon Web Services (AWS)** через платформу **AWS Public Datasets**. Цей сервіс надає доступ до великих наборів даних, які можуть бути корисними для аналізу ринку та прогнозування продажів.

Найбільш підходящим датасетом, що мені вдалось знайти, був на платформі Kaggle. Він містив у собі дані про продажі автомобілів в Америці.
Його переваги:

- Список із більше ніж 500 000 автомобілів, що забезпечить якісне навчання моделі.
- Дані про автомобілі які випускались протягом 33 років.

- Таблиця складається із 16 стовпців, тобто вона містить достатньо критеріїв на яких може навчитись модель.

Його недоліки:

- Дані з американського ринку, що є не актуальним для використання в Україні.
- Найновіший автомобіль в цьому датасеті 2015 року випуску, що є проблемою для прогнозування ціни новіших автомобілів.
- Є стовпець де покупець оцінює стан придбаного автомоблія. Часто ця оцінака не є об'єктивною, що заважає для навчання моделі.

# year The manufacturing year of the vehicle.	make The brand or manufacturer of the vehicle.	model The specific model of the vehicle.	trim dditional designation for the vehicle model.	body The body type of the vehicle (e.g., SUV, Sedan).	transmission The type of transmission in the vehicle (e.g., automatic).	vin Vehicle Identification Number, a unique code for each vehicle.
1982 2015	Ford 17% Chevrolet 11% Other (405086) 72%	Altima 3% F-150 3% Other (525009) 94%	Base 10% SE 8% Other (459372) 82%	Sedan 36% SUV 21% Other (240108) 43%	automatic 85% [null] 12% Other (17570) 3%	550298 unique values
2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472
2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319
2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351
2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987
2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731
2015	Nissan	Altima	2.5 S	Sedan	automatic	1n4al3ap1fn326013
2014	BMW	M5	Base	Sedan	automatic	wbsfv9c51ed593089
2014	Chevrolet	Cruze	1LT	Sedan	automatic	1g1pc5sb2e7128460
2014	Audi	A4	2.0T Premium Plus quattro	Sedan	automatic	wauffafl3en030343
2014	Chevrolet	Camaro	LT	Convertible	automatic	2g1fb3d37e9218789
2014	Audi	A6	3.0T Prestige quattro	Sedan	automatic	wauhgafc0en062916

Рис. 3.1. Фрагмент набору даних

3.2. Навчання та вибір моделі

Перед навчанням моделі я провів певну обробку даних. А саме видалив не важливі стовпці, для текстових полів використав кодування категоріальних змінних, для числових даних – стандартизацію, також замінив відсутні значення на середні значення по колонці.

Для свої роботи я розглядав такі моделі машинного навчання, як лінійна регресія, випадковий ліс та градієнт бустинг. Оскільки лінійна регресія, безпосередньо, використовується для задач регрресії, тобто передбачення числових значень. А випадковий ліс та градієнт бустинг, може

використовуватись як для задач класифікації, так і для задач регресії. Моделі я буду оцінювати такими метриками:

Коефіцієнт детермінації (R^2) — це статистичний показник, який визначає, яку частку дисперсії залежної змінної пояснює незалежна змінна або набір незалежних змінних у моделі регресії. Він варіюється від 0 до 1, де 0 означає, що модель не пояснює взагалі ніякої частки варіації залежної змінної. А 1 означає, що модель повністю пояснює варіацію залежної змінної.

Формула для розрахунку:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Де:

SS_{res} *– сума квадратів залишків (Sum of Squares of Residuals), яка представляє собою різницю між фактичними значеннями та значеннями, передбаченими моделлю.*

SS_{tot} *– загальна сума квадратів (Total Sum of Squares), яка представляє собою різницю між фактичними значеннями та середнім значенням залежної змінної.*

Середньоквадратична похибка (MSE) є мірою якості регресійної моделі. Вона обчислюється як середнє значення квадратів різниць між фактичними та передбаченими значеннями. Чим менше значення MSE, тим краща модель.

Формула для розрахунку MSE:

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y_i})$$

Де:

n *– кількість спостережень.*

y_i *– фактичне значення.*

$\hat{y_i}$ *– передбачене значення.*

Середня абсолютна похибка (MAE) також є мірою якості регресійної моделі. Вона обчислюється як середнє значення абсолютних значень різниць між фактичними та передбаченими значеннями. MAE є більш інтуїтивно зрозумілим показником, оскільки вимірює середню величину помилки в тих самих одиницях, що й самі дані.

Формула для розрахунку MAE:

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}|y_i - \hat{y_i}|$$

Де:

n *– кількість спостережень.*

y_i *– фактичне значення.*

$\hat{y_i}$ *– передбачене значення.*

Лінійна регресія. У своїй роботі я використав множинну лінійну регресію. Оскільки класичний її варіант передбачає знаходження зв'язку між однією незалежною змінною x та залежною змінною y. Модель лінійної регресії має вигляд:

$$y = \beta_0 + \beta_1 x + \epsilon$$

Де:

y – залежна змінна.

x – незалежна змінна.

β_0 – вільний член (інтерсепт), тобто значення y, коли x=0.

β_1 – коефіцієнт нахилу (градієнт), що показує, наскільки зміниться y при зміні x на одну одиницю.

ϵ – випадкова похибка або шум, що враховує вплив інших факторів, не включених у модель.

Множинна лінійна регресія використовує кілька незалежних змінних для прогнозування значення залежної змінної. Модель множинної лінійної регресії має вигляд:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + + \beta_p x_p + \epsilon$$

Де:

y – залежна змінна.

$x_1, x_2, \ldots, x_p$ – незалежні змінні.

β_0 – вільний член (інтерсепт).

$\beta_1, \beta_2, \ldots, \beta_p$ – коефіцієнти при незалежних змінних, що показують, як змінюється y при зміні кожної з незалежних змінних на одну одиницю.

ϵ – випадкова похибка або шум.

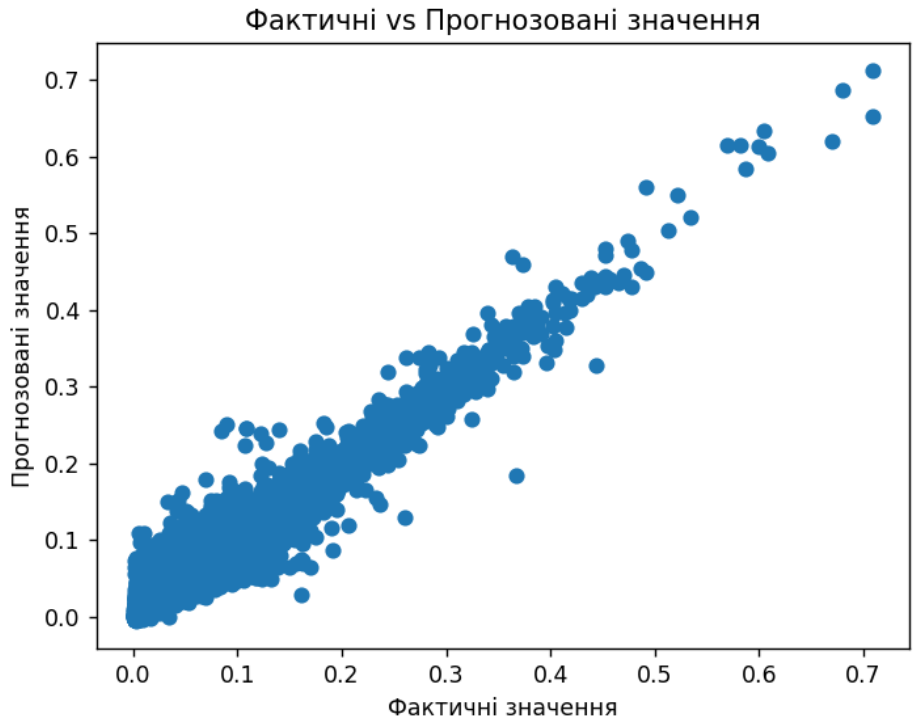

Рис.3.2. графік фактичних і прогнозованих значень для лінійної регресії

Середньоквадратична помилка (MSE): 0.024627881636991155

Коефіцієнт детермінації (R^2): 0.9750172302241433

Середня абсолютна помилка (MAE): 0.1000908019173679

Градієнтний бустинг є потужною технікою ансамблевого навчання, яка об'єднує декілька слабких моделей для створення сильної моделі. В основі градієнтного бустингу лежить ідея побудови нових моделей, які коригують помилки попередніх моделей.

Процес починається з простої моделі, зазвичай константи, яка передбачає середнє значення залежної змінної. Потім на кожній ітерації додається нове дерево рішень, яке навчається на залишках — різниці між фактичними значеннями і передбаченнями поточної моделі. Кожне нове дерево намагається мінімізувати залишки, тобто зробити модель більш точною. Цей процес повторюється, поки не буде додано задану кількість дерев або поки помилки не стануть мінімальними.

Формально, на кожному кроці алгоритм обчислює залишки як негативні градієнти функції втрат. Після цього будується нова модель, яка намагається передбачити ці залишки. Далі обчислюється оптимальний коефіцієнт, який мінімізує функцію втрат при додаванні нового дерева до поточної моделі. Потім модель оновлюється, додаючи до неї нове дерево з відповідним коефіцієнтом.

Для задач регресії градієнтний бустинг зазвичай використовує середньоквадратичну похибку як функцію втрат. На кожному кроці обчислюються залишки між фактичними і передбаченими значеннями, і нове дерево навчається на цих залишках. Остаточна модель складається з суми всіх побудованих дерев.

Щоб спрогнозувати ціни на основі попередніх продажів товарів, потрібно зібрати історичні дані про продажі, підготувати ці дані (заповнити відсутні значення, закодувати категоріальні змінні, стандартизувати числові змінні), розділити їх на тренувальну і тестову вибірки, а потім навчити модель градієнтного бустингу на тренувальних даних. Після навчання модель

оцінюється на тестових даних за допомогою метрик якості, таких як середня абсолютна помилка або середня квадратична помилка. Якщо результати задовольняють, модель використовується для передбачення цін на нових даних.

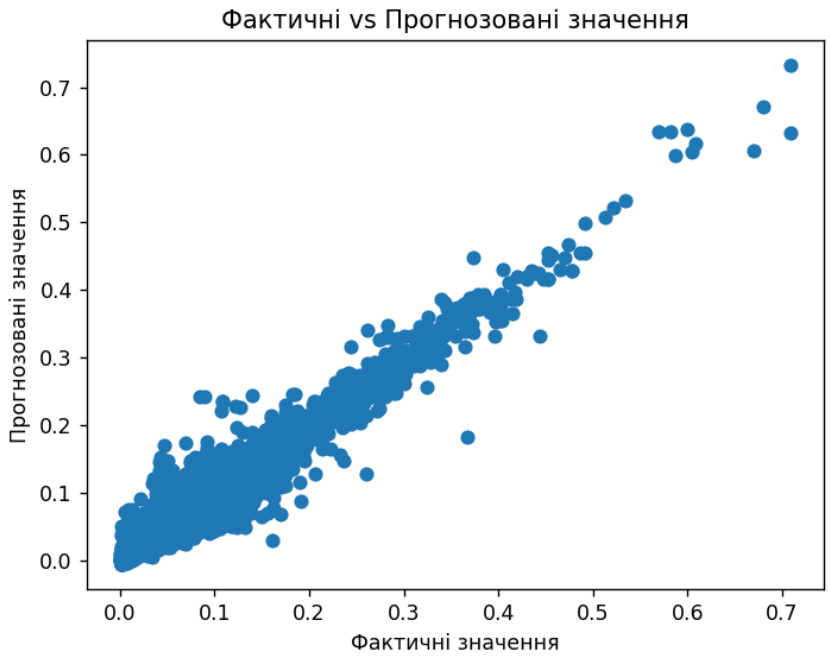

Рис.3.4. графік фактичних і прогнозованих значень для градієнт бустингу

Середньоквадратична помилка (MSE): 4.241884439904719e-05

Коефіцієнт детермінації (R^2): 0.9760519846746554

Середня абсолютна помилка (MAE): 0.0041431815475492725

Випадковий ліс є технікою ансамблевого навчання, яка об'єднує безліч дерев рішень для покращення точності моделі та зниження ризику перенавчання. Ключова ідея випадкового лісу полягає в побудові великої кількості дерев рішень, кожне з яких навчається на різних підмножинах даних і з різними підмножинами ознак. Остаточне передбачення базується на агрегації передбачень усіх дерев: у задачах класифікації використовується голосування більшості, а в задачах регресії — середнє значення.

Алгоритм починається зі збору і підготовки даних, які включають обробку пропущених значень, кодування категоріальних змінних і стандартизацію

числових змінних. Потім створюється ансамбль дерев, кожне з яких навчається на різних підмножинах тренувальних даних, отриманих методом бутстрап-реплікації (random sampling with replacement). Для кожного вузла дерева вибирається випадкова підмножина ознак для визначення найкращого розбиття даних у цьому вузлі.

Навчання кожного дерева рішень відбувається незалежно, використовуючи обрану підмножину даних і ознак. Процес навчання включає поділ даних у вузлах дерева на основі максимізації інформаційного приросту (для задач класифікації) або зменшення дисперсії (для задач регресії). Кожне дерево росте до повної глибини без обрізання, що дозволяє враховувати всі можливі взаємодії між ознаками.

Коли всі дерева побудовані, модель випадкового лісу використовує їх для передбачення нових даних. У задачах класифікації кожне дерево робить передбачення, і остаточне передбачення визначається шляхом голосування більшості серед усіх дерев. У задачах регресії передбачення всіх дерев усереднюються для отримання остаточного результату.

Випадковий ліс має кілька ключових переваг. По-перше, він знижує ризик перенавчання, оскільки кожне дерево навчається на різних підмножинах даних і ознак. По-друге, випадковий ліс здатен ефективно обробляти великі набори даних і велику кількість ознак. По-третє, він забезпечує хорошу узагальнювальну здатність і високу точність передбачень. Однак випадковий ліс може бути обчислювально затратним і вимагати значних обсягів пам'яті, особливо при роботі з великими наборами даних і великою кількістю дерев.

Таким чином, випадковий ліс є потужним і гнучким методом ансамблевого навчання, який широко використовується для задач класифікації та регресії завдяки своїй здатності покращувати точність і знижувати ризик перенавчання.

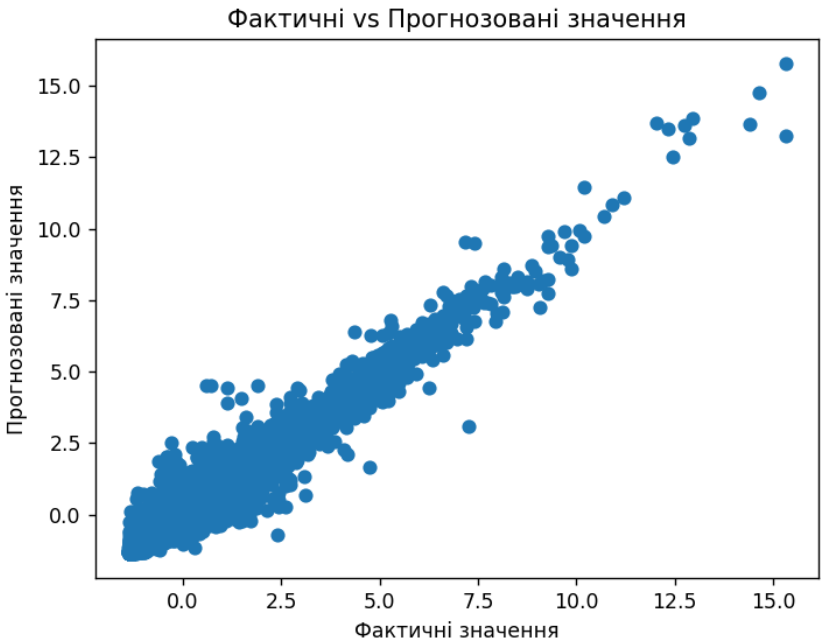

Рис.3.4. графік фактичних і прогнозованих значень для випадкового лісу.

Середньоквадратична помилка (MSE): 0.023608169745702057

Коефіцієнт детермінації (R^2): 0.976051636178877

Середня абсолютна помилка (MAE): 0.09774303502312472

Таким чином, я побачив що всі моделі, показують задовільні результати, але я вибрав модель з випадкового лісу. Оскільки ця модель гарно себе рекомендує в майбутньому перенавчанні. А це мені досить важливо, адже ринок постійно змінюється, і дані слід оновлювати, щоб отримувати вірні прогнози. І ще функціонал перенавчання моделі присутній в моїй програмі.

3.3. Огляд програми

Для розробки інтерфейсу я обрав бібліотеку Flet. Адже вона забезпечує простий і ефективний спосіб створення кроссплатформних інтерфейсів користувача з використанням Python, що дозволяє швидко розробляти та розгортати додатки. Вона пропонує багатий набір віджетів і компонентів, які спрощують процес розробки та зменшують час на створення інтерфейсу.

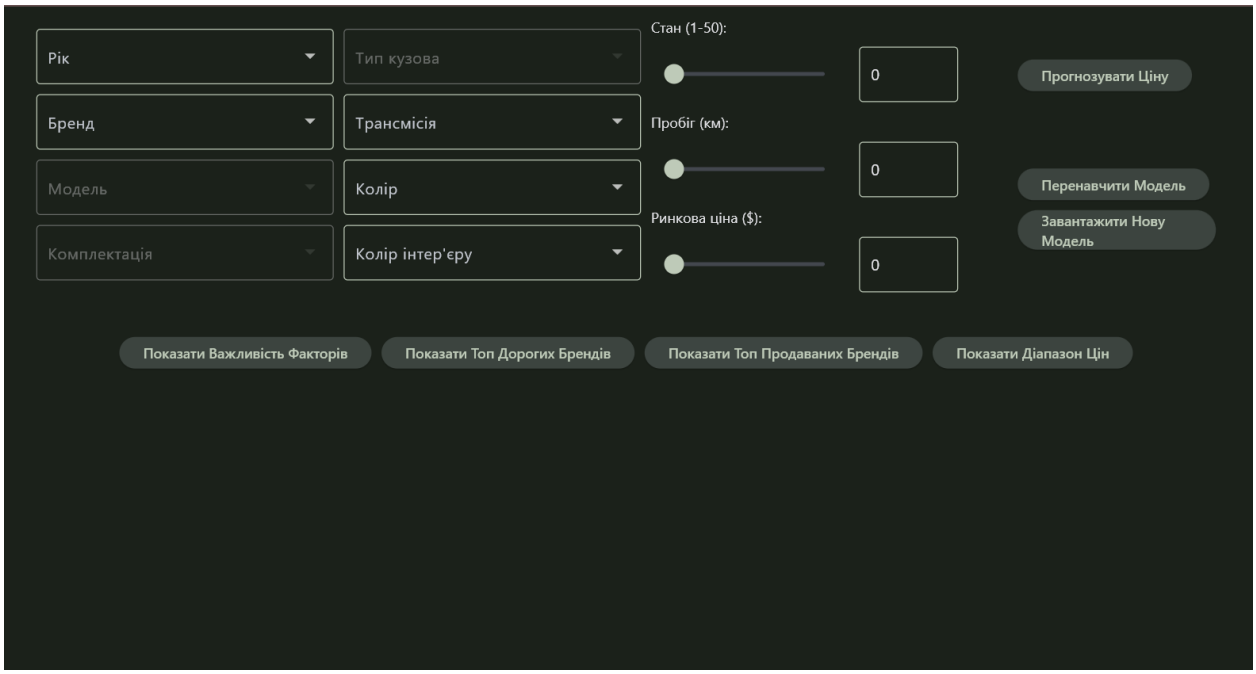

Рис.3.5 Інтерфейс програми

Для зручного користування програмою, введення даних організоване через випадаючі списки, або через повзунки. Цифрові дані, введені через повзунки можуть бути відредаговані з клавіатури. Також організована перевірка, якщо користувач не введе дані, або у випадку цифрових даних, дані що не відповідають умові, поле підсвітиться червоним і прогнозування не буде відбуватись. Також у випадаюих списках поки користувач не вибере бренд, він не зможе вибрати модель, поки він не обере модель, він не зможе вибрати комплектацію і тип кузова. Це зв’язано з тим, щоб користувач не ввів неіснуючий автомобіль, та прогноз відбувався коректно.

ВИСНОВКИ

ДОДАТКИ

Додаток А

Фрагмент написання фронтенду

Додаток В

Фрагмент написання бекенду

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. RUVDS.com. URL: https://habr.com/ru/companies/ruvds/articles/343022/ (дата звернення 14.05.2023).
2. Філософія React. URL: https://uk.legacy.reactjs.org/docs/thinking-in-react.html (дата звернення 20.05.2023).
3. Ant Design of React. URL: https://ant.design/docs/react/introduce/(дата звернення 1.06.2023).
4. Документація .NET. URL: https://learn.microsoft.com/ru-ru/dotnet/core/releases-and-support (дата звернення 28.05.2023).
5. SQLite. URL: https://sqlite.org/index.html (дата звернення 24.05.2023).
6. React.js. URL: https://uk.reactjs.org (дата звернення 20.05.2023).
7. Frontend and Backend. URL: https://en.wikipedia.org/wiki/Front_end_and_back_end (дата звернення 25.05.2023).
8. Database. URL: https://en.wikipedia.org/wiki/Database (дата звернення 05.04.2022).
9. NPM. URL: https://www.npmjs.com (дата звернення 12.05.2023).
10. MongoDB. URL: https://www.mongodb.com (дата звернення 20.05.2023).
11. Інтернет-журналістика та блогінг. URL: http://kzgizh.knukim.edu.ua/entrant/specialty/internet-zhurnalistyka-ta-blohinh (дата звернення 25.05.2023).
12. Mongoose. URL: https://mongoosejs.com/ (дата звернення 16.05.2023)
13. NoSQL: URL: https://www.mongodb.com/nosql-explained (дата звернення 18.05.2023).
14. Passport.js. URL: https://www.passportjs.org (дата звернення 22.05.2023).
15. Redux. URL: https://redux.js.org (дата звернення 24.05.2023).