

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____

(повна назва)

Кафедра _____ програмної інженерії _____

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження методів удосконалення програмної складової систем
транспортної логістики

(тема)

Виконав:

студент 2 курсу, групи ІІЗМ-22-3

_____ Ареш'єв О. О. _____

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник _____ проф. каф. ІІ Шостак І. В. _____

(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

_____ (підпис)

_____ З.В.Дудар _____

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання) _____
 Кафедра _____ програмної інженерії _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 Тип програми _____ освітньо-наукова програма _____
 Освітня програма _____ Інженерія програмного забезпечення _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Ареш'єву Олексію Олександровичу _____

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів удосконалення програмної складової систем транспортної логістики»

Затверджена наказом по університету від 01.11.2023 р.2. Термін подання студентом роботи до екзаменаційної комісії 14.06.2024 р.

3. Вихідні дані до роботи опис систем транспортної логістики, застосування математичних моделей задач, Big Data, аналіз даних за допомогою мови програмування Python, застосування штучного інтелекту

4. Перелік питань, що потрібно опрацювати в роботі

аналіз та порівняння існуючих моделей штучного інтелекту, обрання найбільш підходящої моделі для дослідження, постановка задачі для її вирішення за допомогою математичних моделей, аналіз даних з використанням програмних засобів за допомогою мови програмування Python та штучного інтелекту

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|----|--|--------------------------------|-----------------|
| 1 | Аналіз предметної галузі та постановка задачі | 23.01 – 14.02.24 | <i>виконано</i> |
| 2 | Аналіз та вибір API для дослідження | 15.02 – 24.02.24 | <i>виконано</i> |
| 3 | Аналіз та моделювання предметної області | 17.02 – 28.02.24 | <i>виконано</i> |
| 4 | Планування експериментів | 25.02 – 28.02.24 | <i>виконано</i> |
| 5 | Програмна реалізація кожного з обраних для дослідження API | 25.02 – 01.04.24 | <i>виконано</i> |
| 6 | Експериментальні дослідження | 02.04 – 20.04.24 | <i>виконано</i> |
| 7 | Аналіз результатів експериментальних досліджень та розробка рекомендацій | 20.04 – 23.04.24 | <i>виконано</i> |
| 8 | Написання та оформлення статті та тез доповіді | 17.04 – 23.04.24 | <i>виконано</i> |
| 9 | Підготовка пояснювальної записки | 01.04 – 26.04.24 | <i>виконано</i> |
| 10 | Підготовка презентації та доповіді | 26.04 – 2.05.24 | <i>виконано</i> |
| 11 | Нормоконтроль | 3.06 – 07.06.24 | <i>виконано</i> |
| 12 | Рецензування | 08.06 – 10.06.24 | <i>виконано</i> |
| 13 | Занесення диплома в електронний архів | 10.06.2024 | <i>виконано</i> |
| 14 | Попередній захист | 12.06.2024 | <i>виконано</i> |
| 15 | Допуск до захисту у зав. кафедри | 14.06.2024 | <i>виконано</i> |

Дата видачі завдання 15 листопада 2023р.

Студент _____
(підпис)

Ареф'єв О. О. _____

Керівник роботи _____
(підпис)

проф. каф. ПІ Шостак І. В. _____
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 70 с., 47 рис., 7 табл., 6 джерел.

ТРАНСПОРТНА ЛОГІСТИКА, ПІДХОДИ, РОЗРОБКА, НАБІР ДАНИХ, АНАЛІЗ ДАНИХ, ШТУЧНИЙ ІНТЕЛЕКТ

Об'єкт дослідження – процеси програмної реалізації завдань, що пов'язані із транспортною логістикою.

Предмет дослідження – програмна складова транспортної логістики.

Метою роботи є дослідити шляхи підвищення ефективності функціонування програмних систем транспортної логістики.

Методами розробки та проєктування є аналіз проблемної області транспортної логістики, вирішення задач за допомогою математичних моделей та багатокритеріальних задач, аналіз даних за допомогою мови програмування Python та штучного інтелекту.

У результаті виконаної роботи були досліджені та відібрана система для потенційного покращення програмних систем транспортної логістики, порівняні та проаналізовані методи вирішення задач з аналізу даних, що мають місце при реалізації бізнес процесів з перевезення вантажів.

TRANSPORT LOGISTIC, APPROACHES, DEVELOPMENT, DATA SET, DATA ANALYSIS, ARTIFICIAL INTELLIGENCE

The object of research is the processes of software implementation of tasks related to transport logistics.

The subject of research is the program component of transport logistics.

The purpose of the work is to investigate ways of increasing the efficiency of the functioning of transport logistics software systems.

The methods of development and design are analysis of the problem area of transport logistics, problem solving using mathematical models and multi-criteria

problems, data analysis using the Python programming language and artificial intelligence.

As a result of the work performed, a system was researched and selected for the potential improvement of software systems of transport logistics, methods of solving data analysis problems that occur during the implementation of business processes for cargo transportation were compared and analyzed.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу EIArKhNURE.

Я, Ареф'єв Олексій Олександрович, студент гр. ПЗм-22-3, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів удосконалення програмної складової систем транспортної логістики», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

| | |
|--|----|
| Вступ..... | 9 |
| 1 Аналіз предметної області | 10 |
| 1.1 Загальний опис сфери транспортної логістики..... | 10 |
| 1.2 Основні види програмної складової транспортної логістики | 11 |
| 1.3 Система управління транспортними потоками (TMS)..... | 11 |
| 1.4 Система управління складськими запасами (WMS)..... | 12 |
| 1.5 Система планування ресурсів підприємства (ERP) | 13 |
| 1.6 Система керування ланцюгами поставок (SCM)..... | 14 |
| 2 Постановка задачі..... | 16 |
| 2.1 Проблематика програмної підтримки реалізації бізнес процесів транспортної логістики | 16 |
| 2.2 Технічна складова та порядок виконання дослідження | 16 |
| 3 Підготовка до проведення дослідження..... | 18 |
| 3.1 Вибір систем та критеріїв для багатокритеріальної задачі | 18 |
| 3.2 Вирішення багатокритеріальної задачі | 20 |
| 4 Постановка задачі розподілу ресурсів..... | 26 |
| 4.1 Проблематика | 26 |
| 4.2 Опис задачі | 26 |
| 5 Формування вимог | 27 |
| 5.1 Загальний опис задачі розподілу ресурсів | 27 |
| 5.2 Вимоги до задачі розподілу ресурсів | 28 |
| 6 Математичне та програмне представлення Задачі розподілу ресурсів | 29 |
| 6.1 Загальне формування задачі | 29 |
| 6.2 Математичне представлення задачі..... | 30 |

| | |
|--|----|
| 6.3 Програмне представлення задачі | 33 |
| 7 Використання аналізу даних та ШІ | 37 |
| 7.1 Загальне формування задачі | 37 |
| 7.2 Програмна реалізація | 37 |
| 7.3 Реалізація з застосуванням штучного інтелекту..... | 46 |
| Висновки..... | 52 |
| Перелік джерел посилання | 53 |
| Додаток А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ | 54 |
| Додаток Б Апробація результатів роботи | 55 |
| Додаток В Слайди презентації | 61 |

ВСТУП

Сфера транспортної логістики є важливою для підтримки глобалізованих ланцюгів поставок та ефективності бізнес-процесів. Важливість цієї галузі неухильно зростає у зв'язку з розвитком міжнародної торгівлі та збільшенням обсягів перевезень. В цьому контексті, програмні системи, які використовуються в транспортній логістиці, відіграють ключову роль у забезпеченні продуктивності, точності та ефективності операцій. Однак, з огляду на швидкі технологічні зміни та зростаючі вимоги ринку, існує безперервна потреба у поліпшенні існуючих програмних рішень.

Це дослідження має на меті проаналізувати сучасний стан програмного забезпечення в області транспортної логістики та ідентифікувати потенційні напрямки його удосконалення. Основна увага приділяється виявленню основних викликів, з якими зіштовхуються транспортно-логістичні компанії, та розробці пропозицій щодо покращення ефективності та функціональності програмного забезпечення. Актуальність цього питання обумовлена постійною потребою в оптимізації логістичних процесів для досягнення більшої ефективності, зниження витрат та підвищення рівня задоволення клієнтів.

Дослідження зосереджується на вивченні сучасних практик та інновацій у сфері програмування, які можуть бути впроваджені для поліпшення програмних компонентів систем транспортної логістики. Метою є визначення та аналіз можливих шляхів удосконалення, що сприятимуть збільшенню конкурентоспроможності компаній та відповідності сучасним вимогам ринку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний опис сфери транспортної логістики

Транспортна логістика[1] – це критична та динамічна сфера, що включає планування, виконання та контроль за переміщенням та зберіганням товарів, послуг та інформації від точки відправлення до точки призначення. Вона охоплює управління всіма аспектами ланцюгів поставок від виробництва до споживача, забезпечуючи ефективність та оптимізацію процесів. Ця сфера має вирішальне значення для глобальної економіки, адже вона впливає на швидкість та вартість доставки товарів, відіграючи ключову роль у світовій торгівлі.

У транспортній логістиці велике значення має інтеграція різних видів транспорту: автомобільного, залізничного, морського та повітряного – для створення ефективних мультимодальних ланцюгів доставки. Сучасні технології, такі як системи GPS, Інтернет речей (IoT) та автоматизоване планування, дозволяють точно відстежувати вантажі, прогнозувати затримки та оптимізувати маршрути. Такий підхід сприяє зниженню витрат, підвищенню точності доставки та покращенню задоволеності клієнтів.

Важливою частиною транспортної логістики є складське управління, яке включає зберігання, управління запасами, комплектацію та розподіл товарів. Ефективність цих процесів безпосередньо впливає на загальну продуктивність ланцюга поставок. Автоматизація складських операцій, використання роботів і передових систем управління запасами дозволяє компаніям скоротити витрати та збільшити швидкість обробки замовлень.

Сфера транспортної логістики також відіграє ключову роль у сталому розвитку, оскільки транспортні операції та логістичні процеси мають значний вплив на навколишнє середовище. Зростаюча увага до екологічної стійкості вимагає від галузі впровадження зелених логістичних стратегій, зокрема, оптимізації маршрутів, використання екологічних транспортних засобів, та підвищення загальної ефективності логістичних операцій.

1.2 Основні види програмної складової транспортної логістики

Серед основних компонентів[2] сфери транспортної логістики, які включають у себе програмну реалізацію можна виділити наступні:

- системи управління транспортними потоками (TMS - Transportation Management Systems);
- системи управління складськими запасами (WMS - Warehouse Management Systems);
- системи планування ресурсів підприємства (ERP - Enterprise Resource Planning);
- системи керування ланцюгами поставок (SCM - Supply Chain Management Systems);
- системи GPS та відстеження вантажів;
- аналітичні та оптимізаційні інструменти.

Програмна складова кожного з вищенаведених компонентів відіграє важливу роль у забезпеченні ефективності та гнучкості транспортної логістики, сприяючи покращенню якості обслуговування клієнтів та зниженню загальних витрат на логістику.

1.3 Система управління транспортними потоками (TMS)

Система управління транспортними потоками (TMS) є ключовим компонентом у сфері логістики, спрямована на оптимізацію та ефективне управління всіма аспектами транспортних операцій. Основними цілями TMS є зниження вартості транспортування, підвищення ефективності доставки, забезпечення прозорості та здатність швидко реагувати на зміни умов ринку. Системи TMS включають управління контрактами з перевізниками, планування маршрутів, відстеження вантажів, оптимізацію вантажних навантажень та аналітику витрат.

Сучасні TMS використовують передові технології, включаючи інтеграцію з GPS для відстеження транспорту, аналітичні інструменти для обробки великих

даних, а також штучний інтелект для покращення прийняття рішень. Це включає функції автоматизованого прогнозування, оптимізації маршрутів та аналізу витрат. Використання хмарних рішень дозволяє забезпечити масштабованість та доступність системи з будь-якої точки світу.

Важливим аспектом TMS є її здатність інтегруватися з іншими бізнес-системами, такими як системи управління складськими запасами (WMS), планування ресурсів підприємства (ERP) та системи керування ланцюгами поставок (SCM). Така інтеграція забезпечує обмін даними в реальному часі між різними відділами та партнерами, поліпшуючи загальну ефективність ланцюга поставок.

Подальший розвиток TMS може включати більш глибоку інтеграцію з інтернетом речей (IoT) для збору даних з транспортних засобів та вантажів, впровадження автономних транспортних систем та розширення можливостей аналітики на основі штучного інтелекту. Це відкриє нові шляхи для оптимізації логістичних процесів та надання ще більш ефективних рішень у галузі транспортної логістики.

1.4 Система управління складськими запасами (WMS)

Системи управління складськими запасами є критичним компонентом сучасної логістики, зосередженим на автоматизації та оптимізації складських процесів. WMS надають можливості для більш ефективного управління запасами, підвищення точності обліку товарних запасів, зниження витрат і покращення якості обслуговування клієнтів. Вони управляють усіма аспектами складської діяльності, від прийому товарів, їх розміщення та зберігання, до підготовки та відвантаження замовлень.

Сучасні WMS використовують передові технології, такі як RFID (радіочастотна ідентифікація), автоматизовані системи зберігання та обробки даних, та інтеграцію з технологіями штрих-кодування. Це дозволяє точно відстежувати переміщення товарів всередині складу, автоматизувати процеси підготовки замовлень і забезпечувати швидке реагування на зміни у запасах.

Використання хмарних технологій і мобільних рішень сприяє гнучкості та доступності системи з будь-якої локації.

Інтеграція WMS з іншими ключовими бізнес-системами, як-от системи управління транспортними потоками (TMS) та планування ресурсів підприємства (ERP), є важливою для забезпечення цілісного управління ланцюгом поставок. Така інтеграція дозволяє автоматично обмінюватися даними між різними відділами і партнерами, поліпшуючи координацію, знижуючи ризики помилок і підвищуючи ефективність загальних логістичних процесів.

Використання WMS може суттєво вплинути на операційну ефективність складу, знижуючи час на обробку замовлень та підвищуючи точність виконання. Автоматизація складських процесів знижує потребу в ручній роботі, мінімізує помилки та забезпечує більш швидке обслуговування клієнтів. Ефективне управління запасами сприяє оптимізації використання складських просторів та зменшенню витрат на зберігання.

Майбутній розвиток WMS може включати ще більшу автоматизацію за допомогою роботизованих систем, ширше використання штучного інтелекту для оптимізації складських процесів, інтеграцію з інтернетом речей (IoT) для покращення відстеження та контролю запасів. Такі нововведення забезпечать вищу адаптивність складів до змінних умов ринку та потреб споживачів, підвищуючи загальну ефективність логістичних ланцюгів поставок.

1.5 Система планування ресурсів підприємства (ERP)

Системи ERP призначені для інтегрованого управління основними бізнес-процесами підприємства в режимі реального часу. Вони об'єднують різноманітні функціональні сфери, такі як: облік, управління запасами, закупівлі, продажі, фінанси, HR та інші, в єдину систему. Мета ERP полягає у забезпеченні ефективності, зниженні витрат та поліпшенні оперативності управлінських рішень через централізоване збирання та обробку даних.

Однією з ключових переваг ERP є її здатність інтегрувати різні бізнес-процеси в одну систему, забезпечуючи плавність і послідовність роботи та

усунення ізольованих інформаційних "острівців". Ця інтеграція сприяє автоматизації процесів, зменшенню ручної роботи та помилок, а також забезпечує оперативний доступ до актуальних даних для прийняття рішень.

ERP системи зазвичай мають модульну структуру, що дозволяє підприємствам вибирати необхідні компоненти відповідно до своїх специфічних потреб. Це надає гнучкість при впровадженні та масштабуванні системи, дозволяючи підприємствам розширювати або модифікувати свої ERP рішення з розвитком бізнесу.

ERP системи включають в себе розширені аналітичні інструменти, які дозволяють здійснювати глибокий аналіз даних, виявляти тренди, прогнозувати потреби та оптимізувати ресурси. Використання аналітики в ERP сприяє кращому розумінню бізнес-процесів, ефективному управлінню запасами, плануванню бюджетів та підвищенню загальної продуктивності підприємства.

Попри численні переваги, впровадження ERP може бути складним процесом, що вимагає значних ресурсів, часу та змін у стандартних бізнес-процесах. Майбутнє ERP може бути пов'язане з більшою інтеграцією з хмарними технологіями, штучним інтелектом та машинним навчанням, що забезпечить ще більшу гнучкість, масштабованість та ефективність управління ресурсами підприємства.

1.6 Система керування ланцюгами поставок (SCM)

Системи управління ланцюгами поставок (SCM) відіграють ключову роль у координації та оптимізації всіх етапів ланцюга поставок, від закупівлі сировини до доставки кінцевого продукту споживачу. Вони дозволяють підприємствам планувати, реалізовувати та контролювати потік товарів та інформації, забезпечуючи ефективність, зниження витрат та підвищення якості обслуговування клієнтів. Основна мета SCM – створити прозорий, гнучкий та реактивний ланцюг поставок.

SCM інтегрує різноманітні функціональні області, включаючи управління закупівлями, виробництвом, складськими запасами, транспортуванням та

відносинами з клієнтами. Ця інтеграція забезпечує єдиний потік даних та процесів, що важливо для точного планування та ефективного виконання операцій. Автоматизація допомагає скоротити час на обробку замовлень, знизити помилки та покращити загальну продуктивність.

Сучасні SCM системи використовують передові технології, такі як штучний інтелект, машинне навчання та великі дані для покращення прийняття рішень та оптимізації процесів. Це включає прогнозування попиту, планування ресурсів та управління ризиками. Інтеграція з хмарними технологіями та мобільними додатками забезпечує гнучкість та доступність системи з будь-якого місця.

Впровадження ефективних SCM систем дозволяє компаніям оптимізувати ланцюги поставок, підвищити прозорість та знизити витрати. Це забезпечує кращий контроль над запасами, мінімізацію втрат, підвищення задоволеності клієнтів та здатність швидко реагувати на зміни ринкових умов.

Майбутнє SCM передбачає більшу інтеграцію з технологіями Інтернету речей (IoT) для збору даних безпосередньо з обладнання, автономних транспортних систем для покращення доставки, та застосування розширеної аналітики для прогнозування та оптимізації ланцюга поставок.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Проблематика програмної підтримки реалізації бізнес процесів транспортної логістики

Сфера транспортної логістики постійно розвивається, але не дивлячись на це, існує багато видів робіт, які досі використовують ручну працю, наприклад, для обробки великого об'єму даних, пошуку оптимальних рішень, прогнозування тощо. Окрім цього, у сфері транспортної логістики зростає потреба в інтеграції ІІІ для оптимізації різних процесів, включаючи управління запасами, планування маршрутів, прогнозування попиту та автоматизацію складських операцій. ІІІ може значно підвищити ефективність логістичних операцій, проте його впровадження супроводжується рядом викликів, таких як потреба у великих обсягах даних, інтеграції з існуючими системами та розробки адаптивних алгоритмів.

У даному дослідженні будуть розглянуті шляхи удосконалення програмної складової транспортної логістики, враховуючи тенденції розвитку та потенційного застосування штучного інтелекту.

2.2 Технічна складова та порядок виконання дослідження

Так як головною ціллю магістерського дослідження буде розглядання шляхів удосконалення програмної складової транспортної логістики, то необхідно обрати декілька систем, які зможуть виконати вже існуючу роботу більш ефективно та з меншою затратою ресурсів. Після чого необхідно сформулювати багатокритеріальну задачу прийняття рішень та порівняти кожен з систем між собою та обрати ту, яка найбільше підходить до поставленої задачі. Поставлена задача має бути наведена в тестовому варіанті, наближеному до реальної задачі.

Отже, підсумовуючи вищенаведені ключові пункти технічної складової дослідження можна виділити наступні етапи:

- обрання систем для порівняння;

- опис ключових функцій систем;
- постановка багатокритеріальної задачі для обраних систем;
- вирішення багатокритеріальної задачі;
- обрання найкращої системи для подальшого тестування;
- формування задач, які можна удосконалити за допомогою обраної системи;
- тестування обраної системи на тестових задачах.

Кожен з вищенаведених пунктів має своє власне важливе значення та буде розглянутий впродовж дослідження.

3 ПІДГОТОВКА ДО ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

3.1 Вибір систем та критеріїв для багатокритеріальної задачі

Основною задачею буде аналіз і вибір адекватних засобів штучного інтелекту для покращення різних аспектів транспортної логістики, включаючи управління запасами, планування маршрутів та оптимізацію процесів доставки. Оскільки сучасні алгоритми штучного інтелекту, зокрема нейронні мережі, мають великий потенціал у точному та ефективному аналізі складних даних, багатокритеріальною задачею буде вибір такої нейронної мережі, яка найкраще відповідає вимогам ефективності, швидкості та точності обробки даних у сфері транспортної логістики.

Серед великої кількості наявних нейронних мереж, можемо обрати декілька найпопулярніших на даний момент та порівняти їх відносно таких критеріїв:

- точність даних: визначає, наскільки точно нейронна мережа обробляє та аналізує дані. Це включає здатність правильно ідентифікувати, класифікувати та реагувати на інформацію;
- швидкість обробки: оцінює, як швидко мережа може обробляти вхідні дані та надавати результати;
- складність впровадження: відображає труднощі, пов'язані з інтеграцією та використанням нейронної мережі в існуючі системи або процеси;
- масштабованість: оцінює спроможність системи адаптуватися та ефективно функціонувати при збільшенні обсягу даних або розширенні географічного охоплення;
- вартість: включає оцінку загальних витрат на впровадження та експлуатацію нейронної мережі.

Стосовно нейронних мереж можемо вибрати такі як:

BERT від Google[3] - це передова нейронна мережа, яка ефективно може бути застосована для аналізу даних у сфері транспортної логістики. Завдяки своїй високій точності в аналізі контекстуального значення тексту, BERT може аналізувати звіти про перевезення, логістичні записи, відгуки клієнтів, а також

плани розвитку транспортних систем. Швидкість обробки даних є важливою для своєчасної реакції на зміни у логістичних процесах. Втім, впровадження BERT може бути складним, адже потребує адаптації моделі до специфіки логістичних даних. Масштабованість BERT дозволяє обробляти великі обсяги даних, що є критично для компаній з великим потоком інформації. Вартість впровадження варіюється та може бути високою через великі обчислювальні потреби, на даний момент складає в середньому \$0.002 за 750 символів (1000 токенів).

GPT-4 від OpenAI, як одна з найбільших моделей штучного інтелекту, може бути застосована для створення точних та релевантних аналізів у сфері транспортної логістики. Її точність у генерації та аналізі тексту дуже висока, що може бути корисним для аналізу ринкових тенденцій, логістичних стратегій та доповідей про ефективність. Швидкість обробки даних забезпечує миттєвий аналіз, важливий для оперативного реагування на зміни у логістичних умовах. Однак, при масштабуванні для великих даних, GPT-4 може потребувати додаткових зусиль. Вартість використання[4] залежить від обсягу оброблюваних даних. Середня вартість за 750 символів \$0.01.

TensorFlow та Keras як відкриті платформи для глибинного навчання дозволяють створювати спеціалізовані моделі, адаптовані до конкретних завдань. Точність таких моделей залежить від якості тренувальних даних та розробки алгоритмів. Швидкість обробки даних може варіюватися в залежності від складності моделі. Гнучкість цих платформ дозволяє адаптувати моделі під різні логістичні завдання, хоча вони вимагають певного рівня технічного розуміння для ефективного впровадження. Вартість впровадження залежить від обраної моделі та обсягу використаних обчислювальних ресурсів, хоча ці платформи надають безкоштовний доступ до їх використання, тому вартість впровадження є нульовою.

Facebook's RoBERTa[5], аналогічно до BERT, може бути використана для ефективного аналізу даних у сфері транспортної логістики. Її висока точність у розумінні тексту робить її корисною для аналізу великих обсягів логістичної інформації. Швидкість обробки даних є важливою для оперативного реагування на логістичні виклики. Втім, розробка та впровадження такої моделі може бути

складною та вимагати спеціалізованих знань. Масштабованість RoBERTa дозволяє обробляти великі обсяги даних, що є важливим для багатоаспектного аналізу логістичних процесів. Вартість впровадження є безкоштовною, так як це оптимізований метод попереднього навчання обробки мови.

Yolo[6], хоч і більше орієнтована на обробку зображень, може бути адаптована для аналізу візуальних даних у сфері транспортної логістики, наприклад, для моніторингу складських операцій чи відстеження вантажів. Її точність у виявленні та класифікації об'єктів може бути використана для автоматизації контролю якості та управління складськими запасами. Швидкість обробки даних дозволяє аналізувати відео в реальному часі, що є корисним для моніторингу та контролю складських процесів. Масштабованість Yolo добра для аналізу візуальних даних, хоча може бути обмежена при складніших логістичних задачах. Yolo доступна безкоштовно, але додаткові функції можуть вимагати платних сервісів, зокрема існують сервіси, які використовують цю платформу та беруть плату за використання додаткових функцій.

3.2 Вирішення багатокритеріальної задачі

Багатокритеріальна задача прийняття рішень є важливою частиною аналітичної роботи, особливо в умовах, де необхідно врахувати ряд різних факторів та збалансувати їх взаємодію. Вона вимагає не тільки зіставлення та оцінки різних альтернатив за декількома критеріями, але й здатності ідентифікувати та вирішувати конфлікти між цими критеріями.

У сфері транспортної логістики, багатокритеріальний аналіз допомагає знайти оптимальні рішення, що сприяють підвищенню ефективності, зниженню витрат та покращенню загальної продуктивності.

Цей підхід є особливо корисним в контексті швидкого розвитку технологій та змінних умов ринку, де кожне рішення має значний вплив на успішність логістичних операцій.

Перед тим як занести дані до таблиці, треба, щоб кожен з критеріїв мав свою власну шкалу та тип:

- точність даних – порядкова шкала виражена у відсотках, де 0% - це мінімальна точність, 100% - максимальна точність;
- швидкість обробки, складність впровадження та масштабованість – номінальні шкала, що приймають значення: низька (мінімальне значення), середня, висока, дуже висока (максимальне значення);
- вартість за 750 символів (1000 токенів) – порядкова шкала, що виражена в умовних одиницях від 0 до нескінченності, де 0 – це мінімальне значення.

Занесемо усереднені показники цих нейронних мереж, які були взяті як з офіційних сайтів розробників так і з інших публічних джерел до таблиці 3.1.

Таблиця 3.1 – Середні показники обраних нейронних мереж (таблиця виконана самостійно)

| | Точність даних (%) | Швидкість обробки | Складність впровадження | Масштабованість | Вартість за 750 символів (1000 токенів) |
|---------------------|--------------------|-------------------|-------------------------|-----------------|---|
| BERT | 93.2 | Висока | Висока | Висока | 0.002 |
| GPT-4 | 85.5 | Дуже висока | Середня | Середня | 0.01 |
| TensorFlow та Keras | 70 | Середня | Дуже висока | Висока | 0 |
| RoBERTa | 92.5 | Висока | Висока | Висока | 0 |
| Yolo | 76.5 | Середня | Середня | Низька | 0 |

Оскільки таблиця містить деякі критерії з якісними шкалами, її необхідно переробити під кількісні. Критерії швидкість обробки та масштабованість мають однакову номінальну шкалу, тому будуть мати відповідне однакове оцінювання. Для цього призначимо бали для цих шкал від 0 до 4, де 4 – максимальний бал:

- дуже висока – 4 бали;
- висока – 3 бали;
- середня – 2 бали;

- низька – 1 бал.

Стосовно шкали складності впровадження, кількість балів для неї буде інвертована відносно шкал швидкості обробки та масштабованості, так як чим легше розробити систему – тим менша кількість ресурсів буде задіяна. Тому для шкали складності впровадження будемо мати наступні бали:

- дуже висока – 1 бал;
- висока – 2 бали;
- середня – 3 бали;
- низька – 4 бали.

Занесемо кількісні значення до таблиці 3.2.

Таблиця 3.2 – Кількісні показники обраних нейронних мереж таблиці 3

| | Точність даних (%) | Швидкість обробки | Складність впровадження | Масштабованість | Вартість за 750 символів (1000 токенів) |
|---------------------|--------------------|-------------------|-------------------------|-----------------|---|
| BERT | 93.2 | 3 | 2 | 3 | 0.002 |
| GPT-4 | 85.5 | 4 | 3 | 2 | 0.01 |
| TensorFlow та Keras | 70 | 2 | 1 | 3 | 0 |
| RoBERTa | 92.5 | 3 | 2 | 3 | 0 |
| Yolo | 76.5 | 2 | 3 | 1 | 0 |

Аналізуючи вищенаведену таблицю відносно множини Парето, можна сказати, що кожна з нейронних мереж має хоча б один критерій, який є кращим відносно деяких з альтернатив. Звідси можна вважати, що всі альтернативи є конкурентоспроможними.

Нормування – це процес перетворення різних шкал оцінок до спільного діапазону, що дозволяє ефективно порівнювати та аналізувати дані, які спочатку були виміряні за різними шкалами або мали різні одиниці виміру. Зазвичай, нормування зводить значення оцінок до діапазону від 0 до 1 або від -1 до 1, що спрощує процеси аналізу та прийняття рішень, особливо при багатокритеріальному аналізі. Цей метод широко використовується в статистиці, обробці даних,

машинному навчанні та в дослідницькій діяльності, де потрібно врахувати та порівняти різні типи даних.

Нормування оцінок з урахуванням мінімальних (min) та максимальних (max) значень – це статистичний метод, який використовується для перетворення різних шкал оцінювання в єдину стандартизовану шкалу. Це дозволяє порівнювати та аналізувати дані, які мають різні масштаби або межі.

Тому тепер можемо зробити нормування оцінок з урахуванням min та max. Для цього в кожному з критеріїв необхідно знайти мінімальне та максимальне значення та після чого виконати обчислення значення критерію для обраної альтернативи. У цьому процесі кожне окреме значення (оцінка) перетворюється з використанням формули (3.1):

$$f = \frac{f_{\text{розрахункове}} - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}} \quad (3.1)$$

де f - нормоване (стандартизоване) значення;

$f_{\text{розрахункове}}$ - значення оцінки для певної альтернативи за конкретним критерієм;

f_{min} - мінімальне значення в наборі даних;

f_{max} - максимальне значення в наборі даних;

Формула (3.1) використовується для нормування оцінок у множині Парето. Ця формула дозволяє перетворити оцінки різних альтернатив відносно заданого критерію на нормалізоване значення в діапазоні від 0 до 1.

Такий підхід забезпечує можливість порівняння альтернатив з різними вихідними шкалами оцінювання.

Нормування дозволяє враховувати відносну важливість кожного критерію у загальному аналізі, що сприяє більш точному та об'єктивному порівнянню альтернатив.

Після чого заповнюємо таблицю 3.3 відповідними значеннями з нормованими кількісними показниками.

Таблиця 3.3 – Нормування кількісних показників нейронних мереж таблиці 3

| | Точність даних | Швидкість обробки | Складність впровадження | Масштабованість | Вартість за 750 символів (1000 токенів) |
|---------------------|----------------|-------------------|-------------------------|-----------------|---|
| BERT | 1 | 0.5 | 0.5 | 1 | 0.2 |
| GPT-4 | 0.668 | 1 | 1 | 0.5 | 1 |
| TensorFlow та Keras | 0 | 0 | 0 | 1 | 0 |
| RoBERTa | 0.969 | 0.5 | 0.5 | 1 | 0 |
| Yolo | 0.28 | 0 | 1 | 0 | 0 |
| max значення | 93.2 | 4 | 3 | 3 | 0.01 |
| min значення | 70 | 2 | 1 | 1 | 0 |

Визначимо вагові коефіцієнти для критеріїв пропорційним методом відносно вартості, так як нейронна мережа не буде отримувати та виводити велику кількість даних. Звідси можемо визначити ваговий коефіцієнт вартості як q . Так як система повинна оброблювати дані у режимі реального часу, тому швидкість обробки буде мати коефіцієнт $5q$. На другому місці по важливості буде точність даних тому коефіцієнт буде $4q$. Масштабованість є більш важливою за складність впровадження, так як кількість даних буде великою і треба розробити ефективну систему, тому коефіцієнти будуть $2q$ для складності впровадження і $3q$ для масштабованості відповідно.

Ваги всіх критеріїв разом повинні в сумі складати 1. Тому за властивістю суми вагових коефіцієнтів отримуємо, що $15q$, тобто маємо наступні вагові коефіцієнти:

- точність даних: 0.333;
- швидкість обробки: 0.264;
- складність впровадження: 0.132;
- масштабованість: 0.198;
- вартість за 750 символів: 0.066;

Використаємо лінійну адитивну згортку (див. формулу 3.2) так як маємо оцінки за різними критеріями та кількісні показники.

$$q^* = \max \sum_{j=1}^n a_{ij}^* \quad (3.2)$$

де $a_{ij}^* = \begin{cases} 1, & \text{якщо стратегія краще} \\ 0, & \text{якщо стратегія гірше} \end{cases}$

Опираючись на формулу лінійної адитивної згортки та отримані вагові коефіцієнти маємо наступну таблицю:

Таблиця 3.4 – Результат використання адитивної згортки таблиці 3

| | Точність даних | Швидкість обробки | Складність впровадження | Масштабованість | Вартість за 750 символів (1000 токенів) | Q* |
|----------------------|----------------|-------------------|-------------------------|-----------------|---|-------|
| BERT | 1 | 0.5 | 0.5 | 1 | 0.2 | 0.304 |
| GPT-4 | 0.668 | 1 | 1 | 0.5 | 1 | 0.311 |
| Tensor Flow та Keras | 0 | 0 | 0 | 1 | 0 | 0.079 |
| RoBERTa | 0.969 | 0.5 | 0.5 | 1 | 0 | 0.300 |
| Yolo | 0.28 | 0 | 1 | 0 | 0 | 0.101 |
| β | 0.333 | 0.264 | 0.132 | 0.198 | 0.066 | |
| α | 0.00393 | 0,00378 | 0,00031 | 0,05555 | 0,33333 | |

Виходячи з отриманих результатів, можемо сказати, що GPT-4, отримавши найкращий результат у розмірі 0.311 після проведення згортки, стає найкращим вибором нейронної мережі для використання у магістерському дослідженні. На другому місці після GPT-4 розмістилася система BERT з показником 0.304, але RoBERTa має доволі непоганий показник у 0.300. Виходячи з цього всі ці системи є доволі потужними і мають непоганий потенціал для виконання робіт у галузі транспортної логістики.

4 ПОСТАНОВКА ЗАДАЧІ РОЗПОДІЛУ РЕСУРСІВ

4.1 Проблематика

У сфері транспортної логістики математичні моделі відіграють вирішальну роль у підвищенні ефективності та оптимізації різних процесів. Вони є основним інструментом для аналізу, планування, управління та прийняття рішень, дозволяючи точно прогнозувати потреби в ресурсах, оптимально розподіляти вантажопотоки, мінімізувати витрати на транспортування та скорочувати час доставки. Завдяки застосуванню математичних моделей можливо виявляти та усувати "вузькі місця" в логістичних ланцюгах, підвищувати рівень сервісу та задоволення клієнтів, а також адаптуватися до змінних умов ринку. Незважаючи на потенціал математичних моделей, їх застосування вимагає глибокого аналізу, розробки та тестування для відображення реальних логістичних процесів.

4.2 Опис задачі

Беручи до уваги, що математичні моделі займають одне з провідних місць у транспортній логістиці, то поставлена задача має бути максимально наближена та часто використовувана в реальних умовах. За подібну задачу може бути взята задача розподілу ресурсів.

Отже, можна виділити наступні етапи, які необхідно буде реалізувати:

- постановка шаблонної задачі розподілу ресурсів;
- побудова математичної моделі;
- вирішення задачі розподілу ресурсів;
- аналіз отриманих результатів.

Кожен з вищенаведених пунктів має бути виконаний у ході дослідження.

5 ФОРМУВАННЯ ВИМОГ

5.1 Загальний опис задачі розподілу ресурсів

Задача розподілу ресурсів є однією з ключових проблем у сфері транспортної логістики та управління проектами. Вона полягає в ефективному розподілі обмежених ресурсів між конкуруючими проектами або процесами з метою оптимізації загального показника ефективності, такого як мінімізація витрат або максимізація прибутку.

У контексті транспортної логістики, це може означати оптимізацію використання транспортних засобів, маршрутизацію для мінімізації витрат на паливо та час доставки, або ж балансування навантаження на складах. Задача розподілу ресурсів включає в себе визначення кількості ресурсів, необхідних для кожного завдання, встановлення пріоритетів між завданнями та знаходження оптимального розподілу цих ресурсів для досягнення цілей проекту або операції.

Основними етапами розв'язання задачі розподілу ресурсів є:

- визначення доступних ресурсів: ідентифікація всіх ресурсів, доступних для розподілу, включаючи матеріальні, фінансові, людські та інші;
- аналіз потреб: визначення потреб кожного процесу або проекту в ресурсах;
- планування розподілу ресурсів: розробка стратегії для ефективного розподілу доступних ресурсів між проектами або процесами з урахуванням їхніх цілей та пріоритетів;
- виконання та контроль: реалізація плану розподілу ресурсів та моніторинг його ефективності з можливістю коригування для відповідності змінним умовам.

Задача розподілу ресурсів вимагає врахування багатьох факторів, включаючи обмеження ресурсів, часові рамки, взаємозв'язки між завданнями та змінні умови зовнішнього середовища. Ефективне вирішення цієї задачі забезпечує зниження витрат, підвищення продуктивності та здатності адаптуватися до змін, а також покращення загальної ефективності логістичних і управлінських процесів.

5.2 Вимоги до задачі розподілу ресурсів

Задача розподілу ресурсів повинна містити наступне:

- ресурси – умовні ресурси, які будуть розподілені між різними задачами;
- роботи – задачі, які необхідно виконати, використовуючи наявні ресурси;
- обмеження ресурсів (об'єм наявних ресурсів) – обмеження на максимально доступну кількість ресурсів для виконання поставлених робіт;
- обмеження робіт (об'єм необхідних ресурсів) – обмеження, що виражається у сукупній кількості ресурсів для виконання певної роботи.

Усі вищенаведені пункти повинні мати умовні позначення, так як поставлена задача повинна мати універсальний вигляд і легко масштабуватися за необхідності для визначення показників ефективності (швидкості виконання та точності отриманого результату).

Кінець таблиці 4.1

| I | Ресурси (R) | Роботи, що необхідно виконати | | | | | | Об'єм наявних ресурсів (R) |
|---|--------------------------|-------------------------------|-----------------|-----|-----------------|-----|-----------------|-------------------------------------|
| | | J ₁ | J ₁ | ... | J _j | ... | J _m | |
| M | R _n | C _{n1} | C _{n2} | ... | C _{nj} | ... | C _{nm} | b _n |
| | Об'єм необхідних R | a ₁ | a ₂ | ... | a _j | ... | a _m | |

Потрібно знайти такий розподіл ресурсів за роботами, щоб мінімізувалися сумарні витрати на виконання усіх робіт, або максимізувався одержуваний у підсумку загальний прибуток.

6.2 Математичне представлення задачі

Вирішення поставленої задачі буде за допомогою симплекс-методу. Основа алгоритму – ідея цілеспрямованого перебору вершин багатогранника планів задачі, при якому забезпечується збільшення або зменшення цільової функції.

Алгоритм симплекс методу включає два етапи:

- знаходження початкового опорного плану;
- знаходження оптимального плану або встановлення факту необмеженості цільової функції на множині планів задачі.

Базою опорного плану $X=(x_1, x_2, \dots, x_{n+m})$ називають m лінійно незалежних векторів.

Алгоритм симплекс-методу застосовується тільки для канонічної форми ЗЛП, тому задачу в інших формах потрібно зображати у канонічній формі (4.1).

$$F(x) = \sum_{j=1}^n c_j x_j \rightarrow \text{extremum}, \quad (4.1)$$

$$\sum_{j=1}^n a_{ij}x_j = b_i, \forall i = \overline{1, m},$$

$$x_j \geq 0, \forall j = \overline{1, n},$$

Загальну форму для цієї задачі оберемо наступну:

$$Z(x) = \sum_{j=1}^n c_j x_j \rightarrow \text{extremum},$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \forall i = \overline{k+1, l},$$

Для розв'язку за допомогою симплекс-методу, необхідно шляхом створення нових змінних $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ перетворити систему нерівностей в еквівалентну систему рівнянь (4.2).

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} = b_m \end{cases} \quad (4.2)$$

У векторній формі матиме такий вигляд:

$$P_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{m1} \end{pmatrix}; P_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \dots \\ a_{mn} \end{pmatrix}; P_0 = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}; P_{n+1} = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}; P_{n+m} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \end{pmatrix};$$

$$P_1 x_1 + P_2 x_2 + \dots + P_n x_n + \sum_{j=n+1}^{n+m} x_j = P_0$$

Цільова функція матиме вигляд:

$$Z = \sum_{j=1}^n c_j x_j + 0 \cdot x_{n+1} + 0 \cdot x_{n+2} + \dots + 0 \cdot x_{n+m}$$

Звідси симплекс-таблиця буде наступною:

| | | | | | | | | | | | | |
|-------|----------------------|-------|-----------|----------|----------|-----|----------|-----|----------|-----------|-----|-----------|
| | | C_j | | C_1 | C_2 | ... | C_j | ... | C_n | C_{n+1} | ... | C_{n+m} |
| i | B | C_i | $X=P_0$ | P_1 | P_2 | ... | P_j | ... | P_n | P_{n+1} | ... | 0 |
| 1 | P_{n+1} | 0 | X_{n+1} | x_{11} | x_{12} | ... | x_{1j} | ... | x_{1n} | 1 | ... | 0 |
| 2 | P_{n+2} | 0 | X_{n+2} | x_{21} | x_{22} | ... | x_{2j} | ... | x_{2n} | 0 | ... | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| i | P_{n+i} | 0 | X_{n+j} | x_{i1} | x_{i2} | ... | x_{ij} | ... | x_{in} | 0 | ... | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| m | P_{n+m} | 0 | X_{n+m} | x_{m1} | x_{m2} | ... | x_{mj} | ... | x_{mn} | 0 | ... | 1 |
| $m+1$ | z_j | | | | | | | | | | | |
| $m+2$ | $z_j - c = \Delta_j$ | | | | | | | | | | | |

Отже параметри, що відповідають початковому не виродженому опорному плану $X = (0, \dots, 0, x_{n+1}, x_{n+2}, \dots, x_{n+m})$ з базою $P_{n+1}, P_{n+2}, \dots, P_{n+m}$ будуть такими:

- стовпець B – вектори, що входять до бази;
- стовпець C_i – коефіцієнти при змінних x_i у цільовій функції;
- рядок C_j – коефіцієнти при змінних x_j у цільовій функції;
- стовпець $X = P_0$ – базисні змінні;
- $x_{ij} = a_{ij}$ – елементи матриці умов поставленої задачі;
- z_j – значення цільової функції при опорному плані;
- $z_j - c = \Delta_j$ – цільовий рядок.

Значення цільової функції буде:

$$z_j = \sum_{i=1}^m c_i x_{ij},$$

де $j = \overline{1, n + m}$.

Ґрунтуючись на вищезазначеному симплекс-методі можна зрозуміти, як реалізується його програмна частина.

6.3 Програмне представлення задачі

Програмна частина може бути реалізована на багатьох мовах програмування. У даному випадку оберемо мову Python, так як вона є відносно легкою в розумінні, але при цьому має багатий функціонал. Окремим гарним доповненням є наявність великої кількості бібліотек, які дозволяють розширити функціонал мови Python.

Алгоритм симплекс методу повністю реалізований у бібліотеці SciPy. Scipy — це потужна бібліотека на Python, яка використовується для наукових і технічних обчислень. Вона надає багато ефективних і зручних інтерфейсів для таких завдань, як числова інтеграція та інтерполяція. Зокрема додає значних можливостей до Python, надаючи команди та класи високого рівня для маніпулювання та візуалізації даних.

Розглянемо деякі параметри методу `optimize.linprog(method='simplex')`. Даний метод приймає наступні параметри:

- `c` – коефіцієнти лінійної цільової функції, що необхідно мінімізувати;
- `A_ub` – матриця обмеження нерівності. Кожен рядок `A_ub` визначає коефіцієнти обмеження лінійної нерівності на `x`;
- `b_ub` – вектор обмеження нерівності. Кожен елемент представляє верхню межу відповідного значення `A_ub @ x`;
- `bounds` – послідовність пар (`min`, `max`) для кожного елемента в `x`, що визначає мінімальне та максимальне значення цієї змінної рішення.

У результаті свого виконання, метод, у якості однієї з частини результату, виводить оптимальний базис для цільової функції.

Сформуємо загальну форму математичної моделі ЗЛП для тестової задачі.

$$Z = 10x_1 + 70x_2 + 50x_3 + 100x_4 + 60x_5 \rightarrow \max$$

$$\begin{cases} 2x_1 + 4x_2 + 5x_4 + 3x_5 \leq 500 \\ 3x_2 + 5x_3 + 4x_4 + 2x_5 \leq 250 \\ 8x_1 + 2x_2 + 3x_3 + 4x_5 \leq 300 \\ 4x_1 + 2x_2 + 3x_3 + 2x_4 + 2x_5 \leq 100 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

Введемо додаткові коефіцієнти x_6, x_7, x_8, x_9 .

Звідси лінійна форма має вигляд:

$$Z = 10x_1 + 70x_2 + 50x_3 + 100x_4 + 60x_5 + 0 * (x_6 + x_7 + x_8 + x_9) \rightarrow \max$$

Канонічний вигляд:

$$\begin{cases} 2x_1 + 4x_2 + 5x_4 + 3x_5 + x_6 = 500 \\ 3x_2 + 5x_3 + 4x_4 + 2x_5 + x_7 = 250 \\ 8x_1 + 2x_2 + 3x_3 + 4x_5 + x_8 = 300 \\ 4x_1 + 2x_2 + 3x_3 + 2x_4 + 2x_5 + x_9 = 100 \end{cases}$$

Вирішимо поставлену задачу за допомогою SciPy.

Для початку створимо змінні для відповідних коефіцієнтів та обмежень (див. рис. 4.1)

```
objective_coefficients = [-10, -70, -50, -100, -60]
matrix_coefficients = [[2, 4, 0, 5, 3],
                        [0, 3, 5, 4, 2],
                        [8, 2, 3, 0, 4],
                        [4, 2, 3, 2, 2]]
limits = [500, 250, 300, 100]
coefficient_bounds = [(0, float('inf'))] * 5
```

Рисунок 6.1 – Створення змінних (виконано самостійно)

Кожна зі змінних позначає таке:

- objective_coefficients – коефіцієнти цільової функції;
- matrix_coefficients – матриця з коефіцієнтами обмеження нерівності;
- limits – обмеження ресурсів;
- coefficient_bounds – кількість змінних та їх діапазон min, max. У даному випадку всі 5 змінні мають діапазон $(0, +\infty)$.

Так як `linprog` у працює у режимі мінімізації функції, то значення `objective_coefficients` необхідно інвертувати у негативні, так як у даному випадку необхідно максимізувати функцію.

Використаємо метод `linprog` та отримаємо наступні результати (див. рис. 4.2).

```
linprog(c = objective_coefficients,
        A_ub = matrix_coefficients,
        b_ub = limits,
        bounds = coefficient_bounds,
        method = 'simplex')

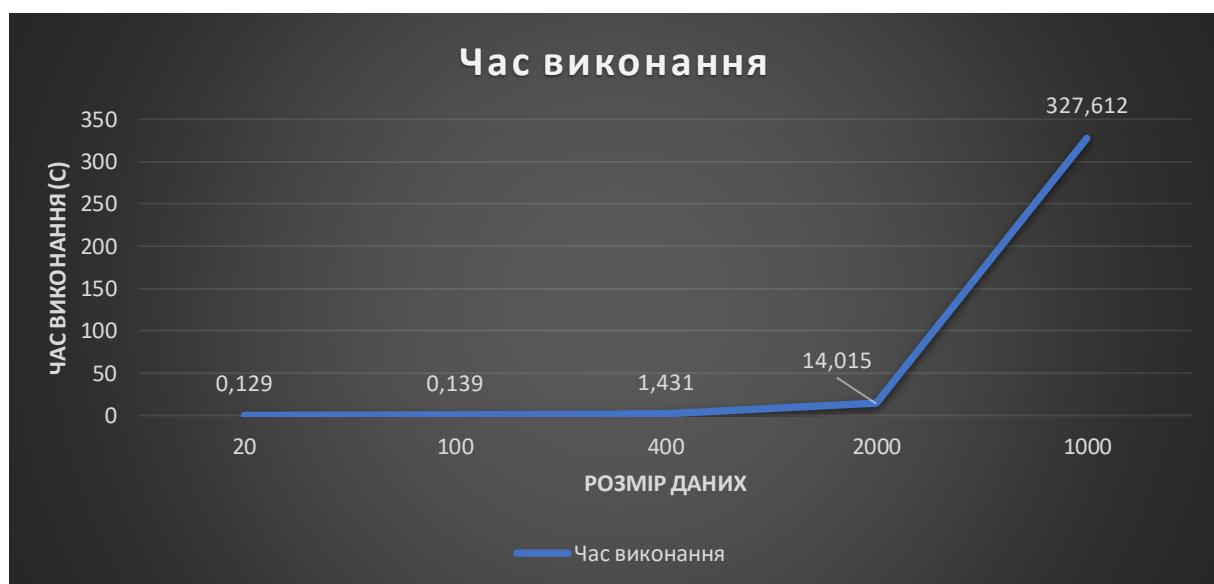
-----
message: Optimization terminated successfully.
success: True
status: 0
fun: -5000.0
x: [ 0.000e+00  0.000e+00  0.000e+00  5.000e+01  0.000e+00]
nit: 7
```

Рисунок 6.2 – Результат виконання методу `linprog`

У результаті видно, що метод відпрацював без помилок. Серед виведеного цікаві лише поля `fun` та `x`. Поле `fun`, результат якого є інвертованим, означає максимум, який можна отримати при коефіцієнтах поля `x`. Тобто при коефіцієнті $x_4 = 5$, буде максимальний результат 5000.

Згенеруємо випадкові дані для 20, 100, 400, 2000, 1000 коефіцієнтів цільової функції та для обмежень і матриці коефіцієнтів. Застосуємо отримані дані для симплекс методу та виміряємо показник часу виконання для кожного з них.

У результаті отримаємо наступний графік:



Отже, з графіку наглядно видно, що зі збільшенням кількості даних значно збільшується час на виконання симплекс-методу.

З великим об'ємом даних часто стикаються великі компанії у логістичних процесах. Саме тому машинне навчання може значно зменшити час обчислень великих даних.

7 ВИКОРИСТАННЯ АНАЛІЗУ ДАНИХ ТА ШІ

7.1 Загальне формування задачі

Аналіз даних та штучний інтелект займають провідну позицію у сучасній сфері транспортної логістики, допомагаючи оптимізувати рух вантажів, покращувати ефективність роботи та знижувати витрати.

На прикладі реальних даних про перевезення вантажів сформуємо типову задачу аналізу:

- виведення таблиці;
- наведення загальних статистичних даних таблиці;
- побудова графіку залежності між різними змінними;
- відсіювання зайвих даних;
- побудова регресійної моделі за заданими параметрами;
- порівняння реальних та прогнозованих даних.

Поставлену задачу вирішимо за допомогою мови програмування Python та відповідних бібліотек, після чого ту саму задачу вирішимо за допомогою штучного інтелекту GPT-4.

7.2 Програмна реалізація

Таблиця FreightRates містить більше 1500 записів та описує всіх доступних кур'єрів, вагу товарів та пов'язані тарифи. Перед початком аналізу таблиці необхідно підключити бібліотеки, які допоможуть проаналізувати дані:

- pandas – використовується для маніпуляції та аналізу даних;
- numpy – використовується для роботи з багатовимірними масивами;
- matplotlib – використовується для візуалізації даних у вигляді графіків та діаграм;
- seaborn – використовується для створення статистичних графіків високої якості.

За допомогою бібліотеки pandas зчитуємо дані таблиці та виведемо перші 5 записів (див. рис. 7.1).

```
freight_rates = pd.read_excel("Supply chain logisitcs problem.xlsx", sheet_name='FreightRates')
freight_rates.head()
```

| | Carrier | orig_port_cd | dest_port_cd | minm_wgh_qty | max_wgh_qty | svc_cd | minimum cost | rate | mode_dsc | tpt_day_cnt | Carrier type |
|---|---------|--------------|--------------|--------------|-------------|--------|--------------|--------|----------|-------------|--------------|
| 0 | V444_6 | PORT08 | PORT09 | 250.0 | 499.99 | DTD | 43.2272 | 0.7132 | AIR | 2 | V88888888_0 |
| 1 | V444_6 | PORT08 | PORT09 | 65.0 | 69.99 | DTD | 43.2272 | 0.7512 | AIR | 2 | V88888888_0 |
| 2 | V444_6 | PORT08 | PORT09 | 60.0 | 64.99 | DTD | 43.2272 | 0.7892 | AIR | 2 | V88888888_0 |
| 3 | V444_6 | PORT08 | PORT09 | 50.0 | 54.99 | DTD | 43.2272 | 0.8272 | AIR | 2 | V88888888_0 |
| 4 | V444_6 | PORT08 | PORT09 | 35.0 | 39.99 | DTD | 43.2272 | 1.0552 | AIR | 2 | V88888888_0 |

Рисунок 7.1 – Виведення даних таблиці FreightRates (виконано самостійно)

Тепер виведемо статистичні дані, що включають: кількість, середнє значення, стандартне відхилення, мінімум, 25-й перцентиль, медіану (50-й перцентиль), 75-й перцентиль і максимальні значення для кожного числового стовпця (див. рис. 7.2).

```
freight_rates.describe()
```

| | minm_wgh_qty | max_wgh_qty | minimum cost | rate | tpt_day_cnt |
|--------------|--------------|--------------|--------------|-------------|-------------|
| count | 1540.000000 | 1540.000000 | 1540.000000 | 1540.000000 | 1540.000000 |
| mean | 155.886637 | 4636.143633 | 12.329033 | 2.892656 | 2.186364 |
| std | 475.952049 | 20251.309727 | 22.946192 | 4.603877 | 2.000895 |
| min | 0.000000 | 0.453592 | 1.202000 | 0.033200 | 0.000000 |
| 25% | 15.010000 | 21.500000 | 3.656800 | 0.470400 | 1.000000 |
| 50% | 41.010000 | 47.500000 | 7.417000 | 1.661200 | 2.000000 |
| 75% | 67.229614 | 99.990000 | 11.494500 | 3.932200 | 2.000000 |
| max | 10000.000000 | 99999.990000 | 425.027200 | 128.027200 | 14.000000 |

Рисунок 7.2 - Загальні статистичні дані (виконано самостійно)

За допомогою бібліотеки seaborn побудуємо графік залежності між різними змінними з таблиці (див. рис. 7.3).

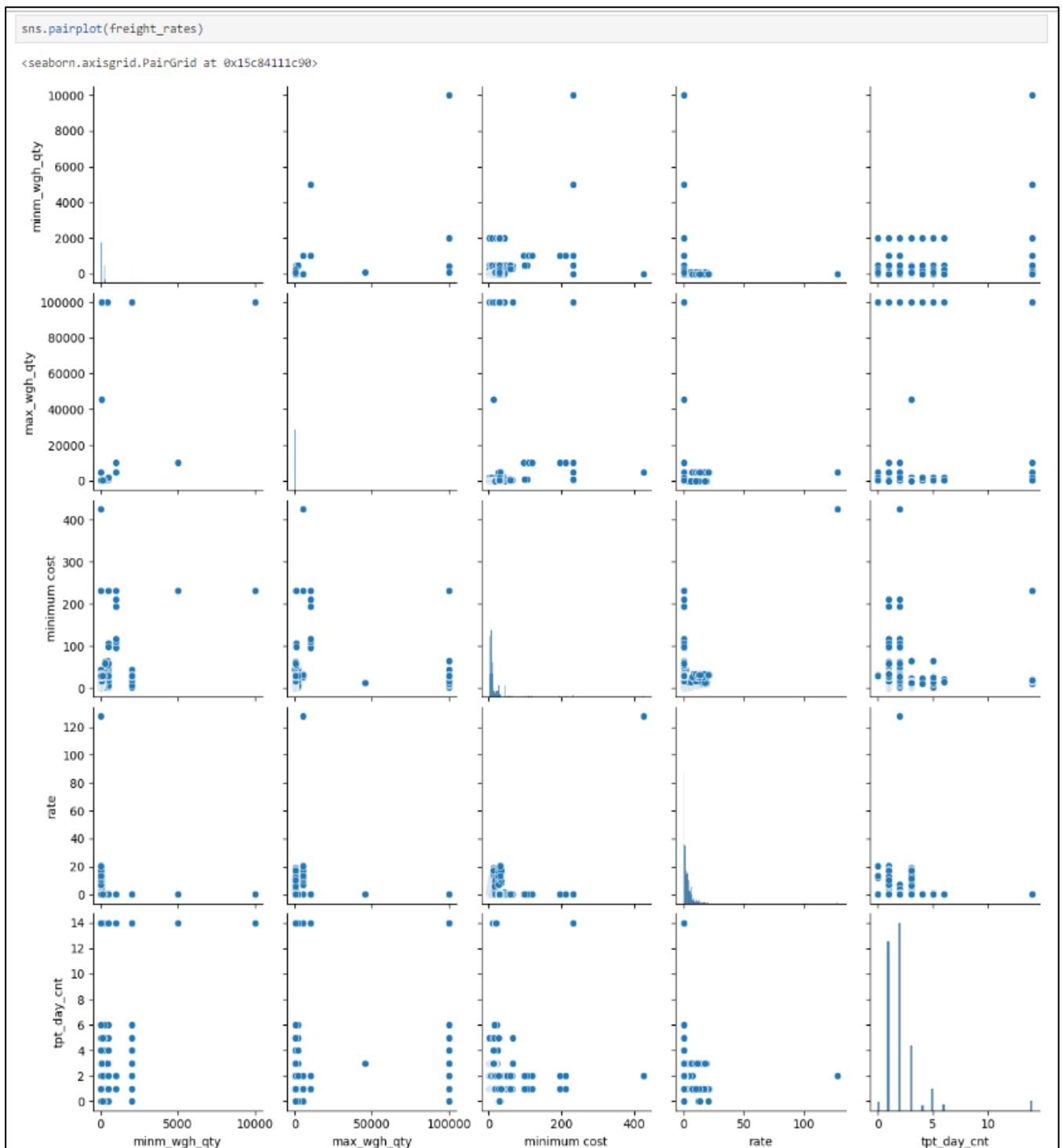


Рисунок 7.3 – Графіки залежності між змінними (виконано самостійно)

З наведених статистичних даних та графіку-порівняння видно, що дані мають outliers, або викиди в даних - це точки або значення в наборі даних, які значно відрізняються від інших спостережень. Вони можуть бути результатом помилок у вимірюванні, збоїв у обладнанні, аномалій у процесах або просто природної варіативності в даних. Викиди можуть суттєво впливати на статистичний аналіз і результати моделювання, тому необхідно їх позбутися.

За допомогою методу міжквартильного розмаху (IQR, Interquartile Range) для виявлення викидів (outliers), який базується на статистичному підході та використовує квартилі даних. IQR — це різниця між третім квартилем (Q3) і першим квартилем (Q1) даних, що представляє середні 50% значень у наборі даних.

Викиди визначаються як точки даних, які лежать за межами наступних порогів:

- нижній поріг: $Q_1 - 1,5 \cdot IQR$;
- верхній поріг: $Q_3 + 1,5 \cdot IQR$.

Значення, які виходять за ці межі, вважаються викидами. Цей метод ефективний для ідентифікації екстремальних значень і часто використовується перед подальшим аналізом даних для підвищення його точності.

Створимо функцію `calc_iqr_limits`, за допомогою якої будуть обраховуватись значення верхнього та нижнього порогів відповідно (див. рис. 7.4).

```
def calc_iqr_limits(column_name):
    Q1 = freight_rates[column_name].quantile(0.25)
    Q3 = freight_rates[column_name].quantile(0.75)

    IQR = Q3 - Q1

    lower_limit = Q1 - 1.5 * IQR
    upper_limit = Q3 + 1.5 * IQR

    return lower_limit, upper_limit
```

Рисунок 7.4 – Функція `calc_iqr_limits` (виконано самостійно)

Після чого застосуємо цю функцію для колонок таких як: `tpt_day_cnt` (кількість днів транспортування), `minimum cost` (мінімальна вартість), `minm_wgh_qty` (кількість товарів з мінімальною вагою), `max_wgh_qty` (кількість товарів з максимальною вагою), що мають викиди та побачимо, що у результаті видалено близько 400 записів.

Відобразимо дані на графіках залежності між змінними (див. рис. 7.5).

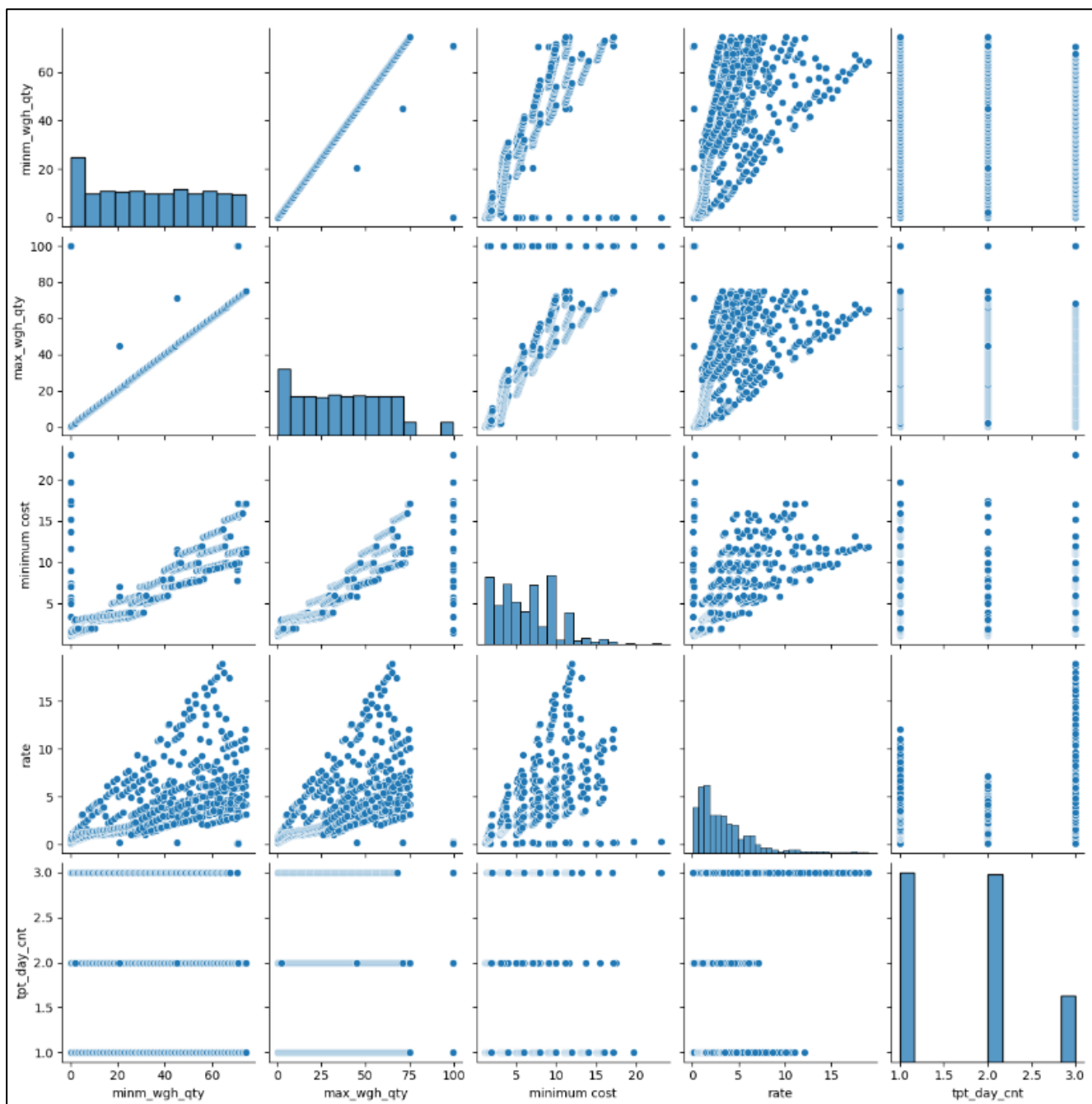


Рисунок 7.5 – Графіки залежності між змінними після застосування IQR
(виконано самостійно)

На вищенаведених графіках видно, що залежності між змінними тепер відображаються більш явно. Таким чином на деяких графіках залежності, як от: мінімальна та максимальна кількість вантажу і мінімальна вартість та тариф (rate) можна побачити, що щодо них можна застосувати лінійну регресію. Вона допоможе спрогнозувати, наприклад, мінімальну ціну для певної кількості вантажу, що треба перевезти.

Виведемо гістограму для мінімальної ціни (minimal cost) (див. рис. 7.6), на якому видно, що основна маса даних варіюється в діапазоні від 2 до 17.

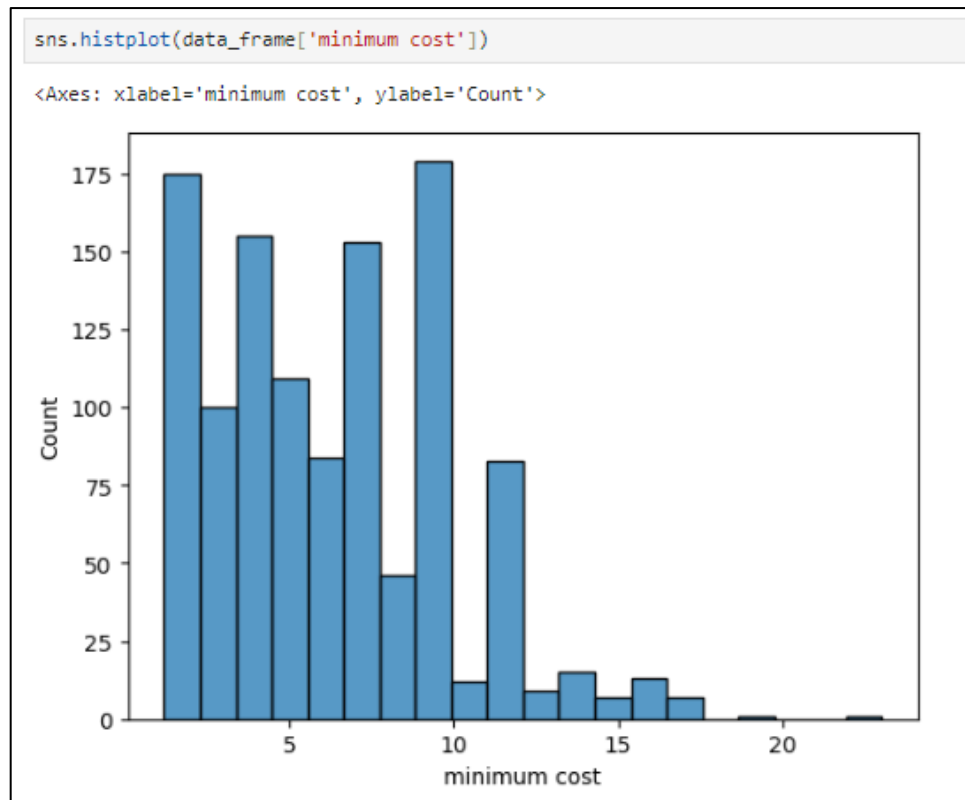


Рисунок 7.6 – Гістограма мінімальної ціни (виконано самостійно)

Надалі необхідно створити регресійну модель. Для цього застосуємо бібліотеку scikit-learn. Scikit-learn – це безкоштовна бібліотека машинного навчання з відкритим кодом для мови програмування Python.

Тепер почнемо тренувати регресійну модель. Спершу нам потрібно буде розділити дані на масив «X», який містить функції для навчання, і масив «y» із цільовою змінною, у даному випадку це стовпець minimal cost. Відкинемо деякі стовпці, як от: order id, product id та декілька інших (див. рис. 7.7) оскільки вони містять лише текстову інформацію, яку модель лінійної регресії не може використовувати.

```
X = data_frame[['minm_wgh_qty', 'max_wgh_qty', 'rate', 'tpt_day_cnt']]
y = data_frame['minimum cost']
```

Рисунок 7.7 – Масиви X та y (виконано самостійно)

Тепер розділимо дані на набір для навчання (X_{train} , y_{train}) та набір для тестування (X_{test} , y_{test}), застосувавши метод `train_test_split` бібліотеки `scikit-learn`. Навчимо модель на «train» наборі (див. рис. 7.8).

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)

from sklearn.linear_model import LinearRegression

lm = LinearRegression()
lm.fit(X_train, y_train)

LinearRegression()
```

Рисунок 7.8 – Навчальні та тестові набори (виконано самостійно)

Оцінимо модель (див. рис. 7.9), перевіривши її коефіцієнти та те, як можемо їх інтерпретувати.

```
coeff_df = pd.DataFrame(lm.coef_, X.columns, columns=['Coefficient'])
coeff_df
```

| | Coefficient |
|---------------------|-------------|
| minm_wgh_qty | 0.042745 |
| max_wgh_qty | 0.077994 |
| rate | 0.296855 |
| tpt_day_cnt | -0.396625 |

Рисунок 7.9 – Коефіцієнти отриманої регресійної моделі (виконано самостійно)

Отже інтерпретація коефіцієнтів при утриманні значень всіх стовпців та збільшені значення на одну одиницю стовпця:

- `minm_wgh_qty` – призводить до збільшення мінімальної ціни на 0.042745;
- `max_wgh_qty` – призводить до збільшення мінімальної ціни на 0.077994;
- `rate` – призводить до збільшення мінімальної ціни на 0.296855;
- `tpt_day_cnt` – призводить до зменшення мінімальної ціни на 0.39662.

Тобто збільшення тарифу перевезення (`rate`) має найбільший вплив на мінімальну ціну, збільшуючи її на майже 0.3 за кожну одиницю. У той час більша кількість днів доставки (`tpt_day_cnt`) призводить до зменшення мінімальної ціни на

майже 0.4 за кожну одиницю, тобто, наприклад, експрес доставка у декілька днів буде коштувати значно більше чим доставка у той самий пункт .

Відобразимо графік залежності реальної (actual) та прогнозованої (predicted) мінімальної ціни (див. рис. 7.10).

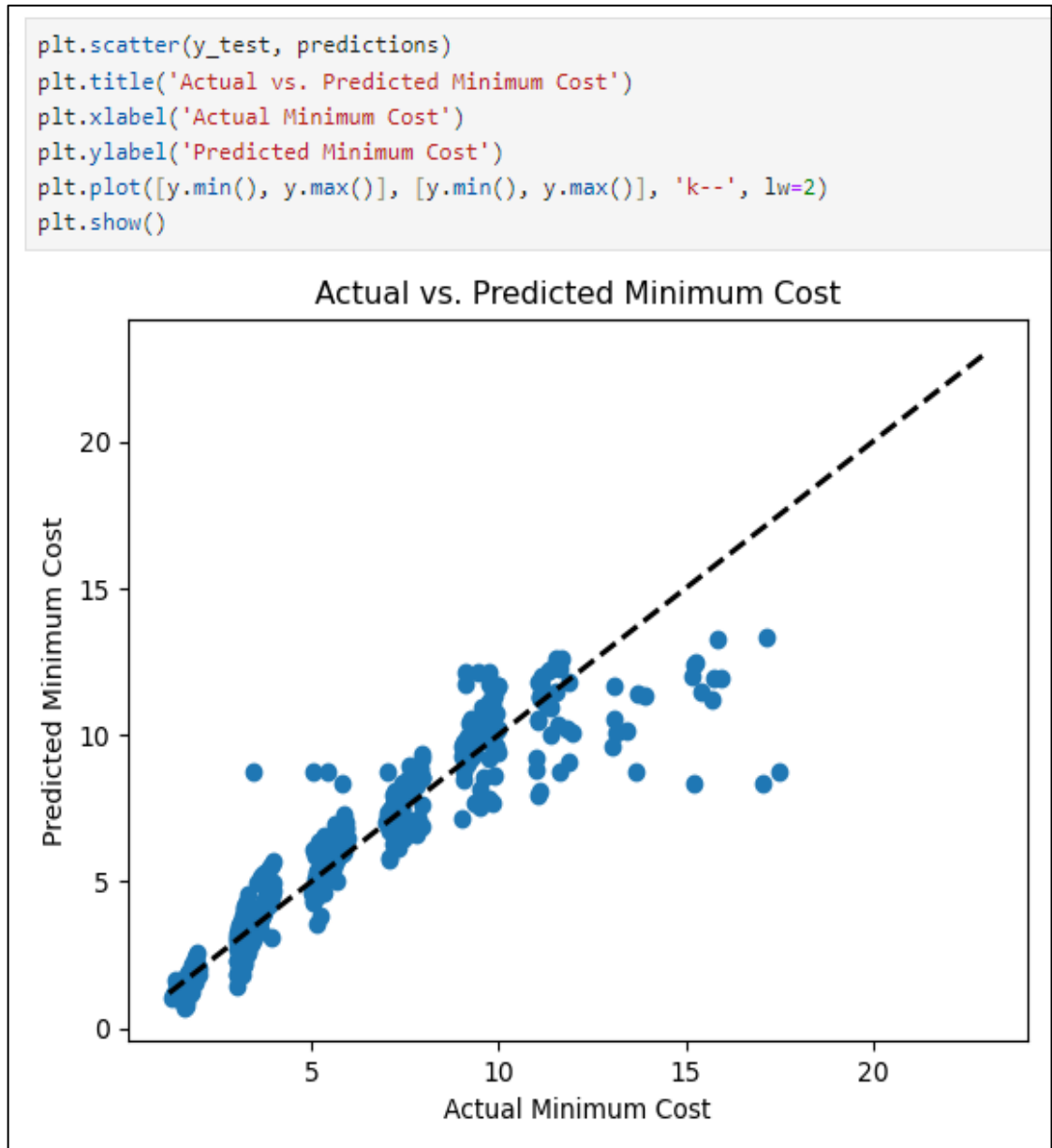


Рисунок 7.10 – Графік залежності реальної (actual) та прогнозованої (predicted) мінімальної ціни (виконано самостійно)

На графіку чорною пунктирною лінією позначено де реальні та прогнозовані дані співпадають. У той час точки, що далі від цієї лінії, означають похибку в прогнозованих даних.

Оцінімо отриману регресійну модель за допомогою метрик:

- середня абсолютна похибка (MAE) – це середнє абсолютне значення похибок, що визначається за формулою

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

- середня квадратична похибка (MSE) – це середнє квадратичних помилок, що визначається за формулою

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

- середньоквадратична похибка (RMSE) – це квадратний корінь із значення середньо квадратичної помилки, що визначається за формулою

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

За допомогою бібліотеки sklearn виведемо значення цих метрик для тестових та прогнозованих даних (див. рис. 7.11)

```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

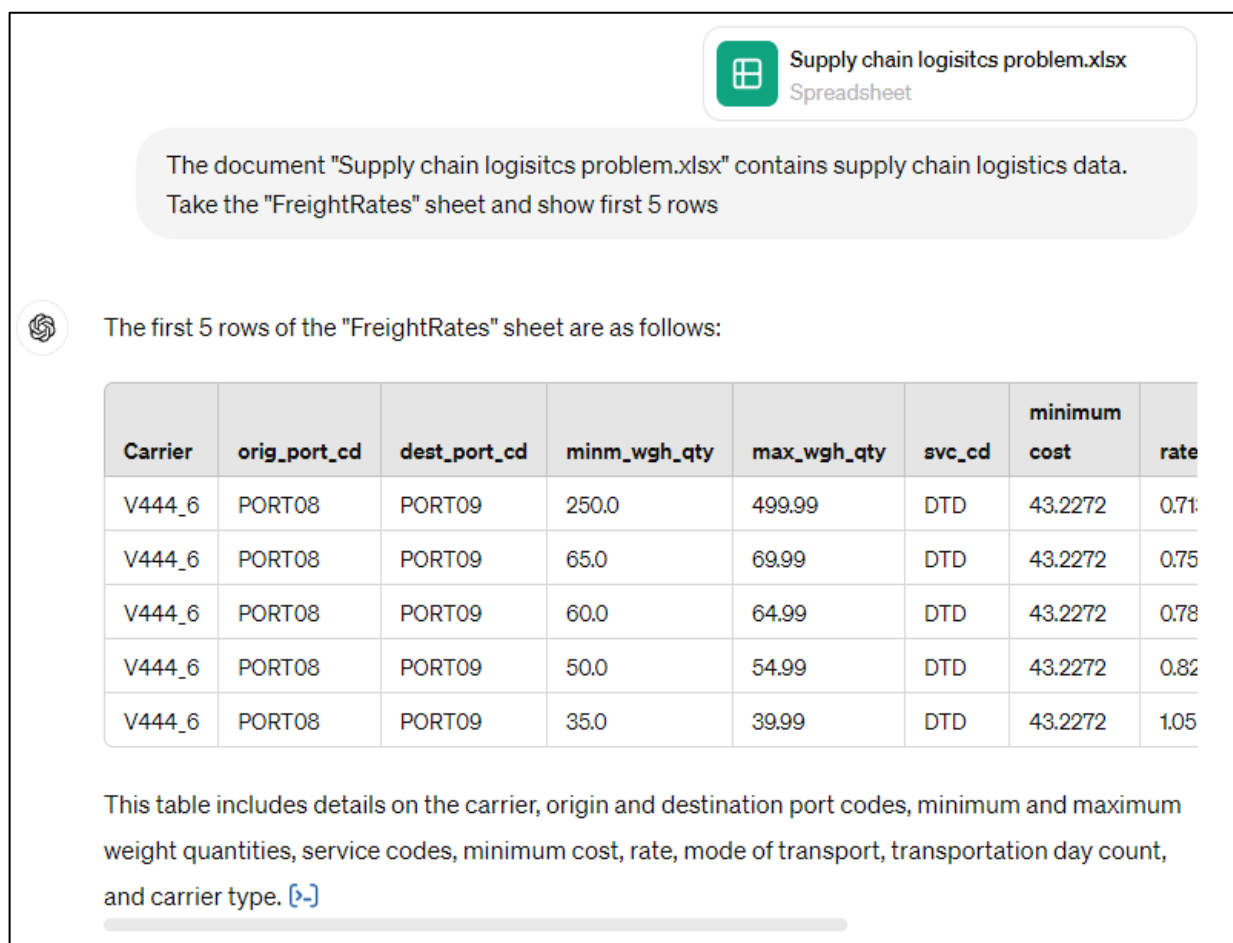
MAE: 0.8060278754580981
MSE: 1.6399340915369338
RMSE: 1.2805991142964819
```

Рисунок 7.11 – Значення метрик (виконано самостійно)

З результатів видно, що всі значення метрик не є завищеними та знаходяться в межах до 1.6, що означає дуже непоганий результат для створеної моделі лінійної регресії для обраних даних.

7.3 Реалізація з застосуванням штучного інтелекту

На даний момент часу, версія GPT-4 дозволяє прикріплювати та аналізувати документи різних форматів. Отже для того, щоб зробити подібний аналіз даних як і в попередньому пункті необхідно завантажити документ та описати, що треба з ним зробити. Для початку вкажемо у запиті, що необхідно вивести перші 5 рядків з таблиці «FreightRates» (див. рис. 7.12).



Supply chain logisitcs problem.xlsx
Spreadsheet

The document "Supply chain logisitcs problem.xlsx" contains supply chain logistics data. Take the "FreightRates" sheet and show first 5 rows

The first 5 rows of the "FreightRates" sheet are as follows:

| Carrier | orig_port_cd | dest_port_cd | minm_wgh_qty | max_wgh_qty | svc_cd | minimum cost | rate |
|---------|--------------|--------------|--------------|-------------|--------|--------------|------|
| V444_6 | PORT08 | PORT09 | 250.0 | 499.99 | DTD | 43.2272 | 0.71 |
| V444_6 | PORT08 | PORT09 | 65.0 | 69.99 | DTD | 43.2272 | 0.75 |
| V444_6 | PORT08 | PORT09 | 60.0 | 64.99 | DTD | 43.2272 | 0.78 |
| V444_6 | PORT08 | PORT09 | 50.0 | 54.99 | DTD | 43.2272 | 0.82 |
| V444_6 | PORT08 | PORT09 | 35.0 | 39.99 | DTD | 43.2272 | 1.05 |

This table includes details on the carrier, origin and destination port codes, minimum and maximum weight quantities, service codes, minimum cost, rate, mode of transport, transportation day count, and carrier type. [↔]

Рисунок 7.12 – Виведення перших 5 рядків таблиці «FreightRates» (виконано самостійно)

Так як дані таблиці були успішно зчитані можна формувати наступний запит. Тепер необхідно вивести загальні статистичні дані такі як: кількість (count), середнє значення (mean), стандартне відхилення (std), мінімум (min), 25-й центиль (25%), медіану (50-й центиль) (50%), 75-й центиль (75%) і максимальні значення для кожного числового стовпця (max). У запиті вкажемо, щоб отримані дані були оформлені в вигляді таблиці (див. рис. 7.13).

Perform next tasks I provide for FreightRates sheet

Give next information about number values for each column: count, mean, std, min, 25%, 50%, 75%, max.

Provide this information as table.

< 2/2 >

Here are the descriptive statistics for the numerical columns in the "FreightRates" sheet:

| Column | Count | Mean | Std | Min | 25% | 50% | 75% | Max |
|--------------|-------|---------|----------|------|-------|-------|-------|----------|
| minm_wgh_qty | 1540 | 155.89 | 475.95 | 0.00 | 15.01 | 41.01 | 67.23 | 10000.00 |
| max_wgh_qty | 1540 | 4636.14 | 20251.31 | 0.45 | 21.50 | 47.50 | 99.99 | 99999.99 |
| minimum cost | 1540 | 12.33 | 22.95 | 1.20 | 3.66 | 7.42 | 11.49 | 425.03 |
| rate | 1540 | 2.89 | 4.60 | 0.03 | 0.47 | 1.66 | 3.93 | 128.03 |
| tpt_day_cnt | 1540 | 2.19 | 2.00 | 0.00 | 1.00 | 2.00 | 2.00 | 14.00 |

These statistics provide a comprehensive overview of the numerical data within the "FreightRates" sheet, including the count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for each numerical column. [\[-\]](#)

Рисунок 7.13 – Загальні статистичні дані (виконано самостійно)

Надані дані містять таку саму інформацію як і в попередньому пункті з програмною реалізацією.

Отже таблиця містить викиди в даних (outliers), яких необхідно позбутись. Запитаємо систему, щоб вона позбулася цих даних, але при цьому не будемо вказувати в запиті метод та яким чином вона повинна це зробити (див. рис. 7.14).

Detect outliers and remove them

Before removing outliers, the "FreightRates" sheet contained 1,540 rows. After outlier detection and removal using the Interquartile Range (IQR) method, 1,089 rows remain. This process removed outliers across all numerical columns in the dataset. [\[-\]](#)

Рисунок 7.14 – Видалення викидів з даних (виконано самостійно)

З відповіді видно, що модель видалила близько 500 записів, використовуючи метод міжквартильного розмаху (IQR) так само як і в програмній реалізації.

Сформуємо запит на виведення порівняльного графіку залежності між колонками для даних без викидів (див. рис. 7.15).

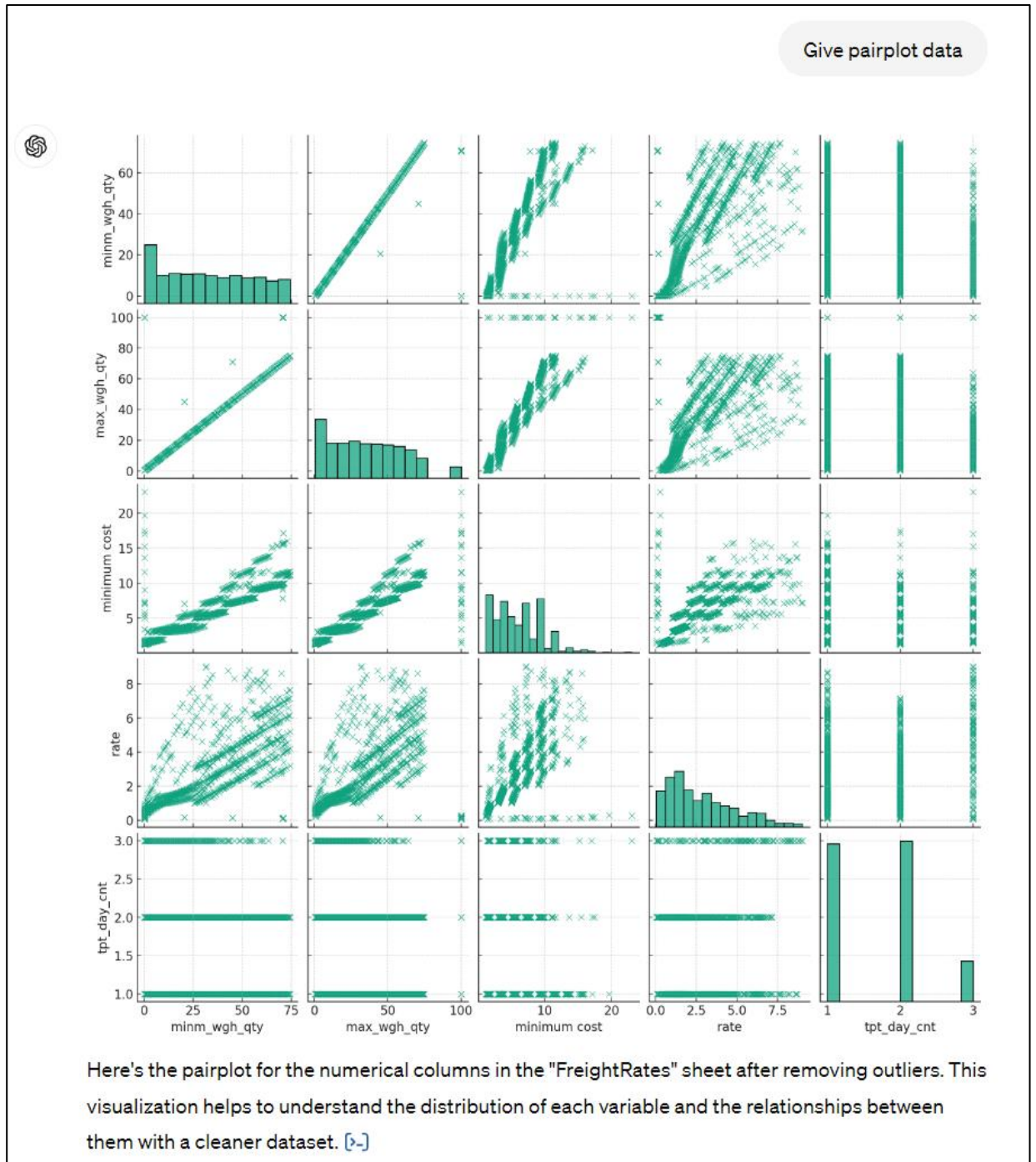


Рисунок 7.15 – Графік залежності між колонками (виконано самостійно)

На отриманому графіку видно, що модель дійсно прибрала зайві дані та залежність між різними колонками тепер більш чітко проглядається.

Виведемо гістограму мінімальної ціни (minimum cost), відносно якої буде побудована регресійна модель (див. рис. 7.16).

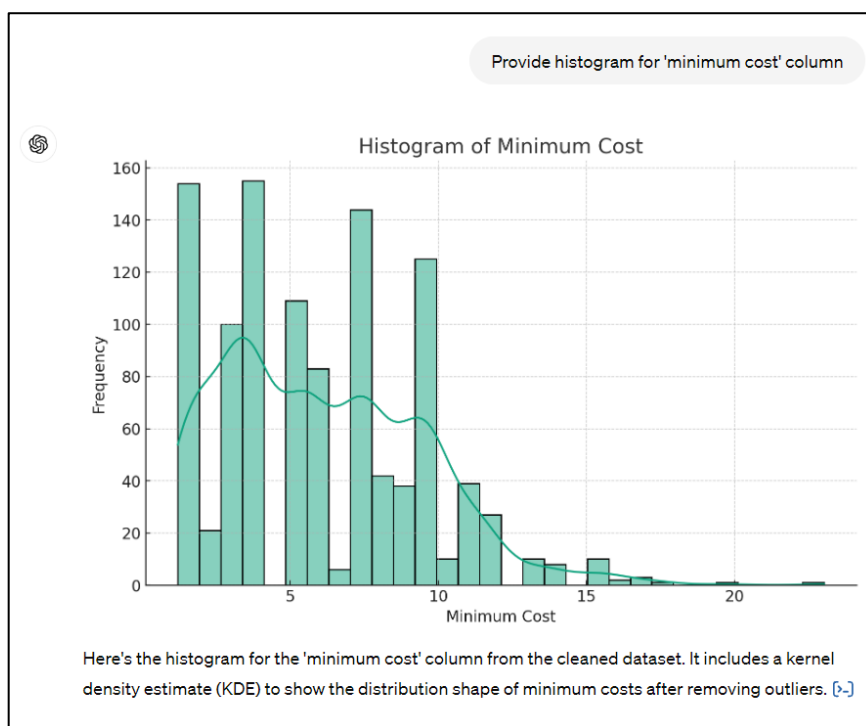


Рисунок 7.16 – Гістограма мінімальної ціни без викидів в даних (виконано самостійно)

Тепер сформуємо запит на створення лінійної регресійної моделі. Модель так само буде тренуватися на колонках: `minm_wgh_qty`, `max_wgh_qty`, `rate`, `tpt_day_cnt`, та вивести метрики: середня абсолютна похибка (MAE), середня квадратична похибка (MSE), середньоквадратична похибка (RMSE) (див. рис. 7.17).

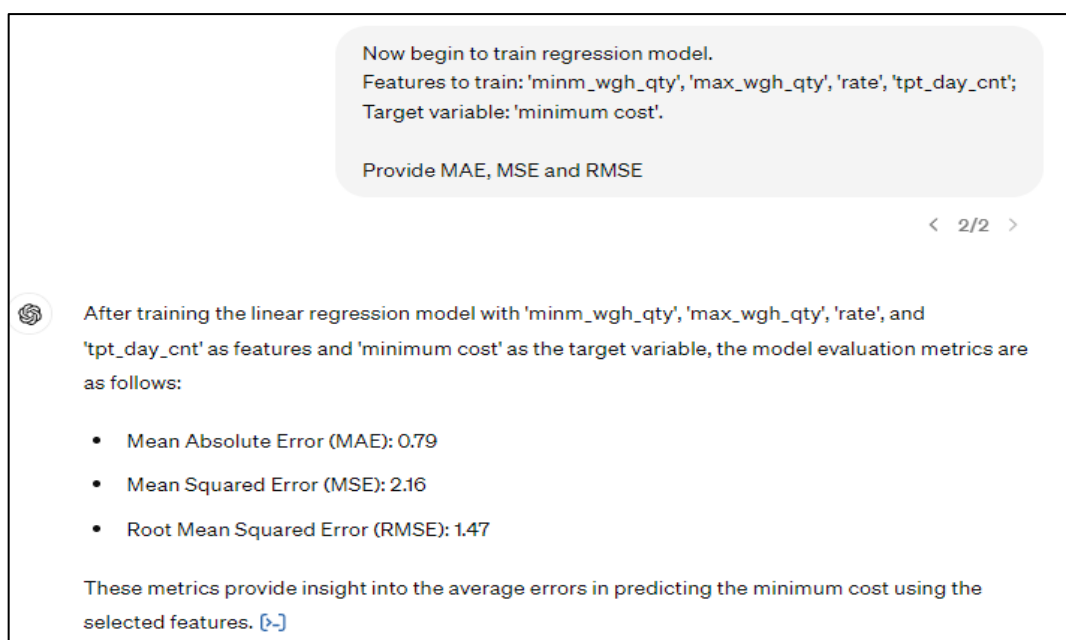


Рисунок 7.17 - Отримані метрики лінійної регресійної моделі (виконано самостійно)

Отримані метрики порівняємо з програмним підходом до задачі:

| Метрика | Програмна реалізація | GPT-4 | Різниця показників |
|---------|----------------------|-------|--------------------|
| MAE | 0,86 | 0,79 | 0,07 |
| MSE | 1,63 | 2,16 | 0,53 |
| RMSE | 1,28 | 1,47 | 0,19 |

З отриманих даних видно, що різниця показників не занадто велика, а отже модель GPT-4 впоралася не гірше за програмну реалізацію, вважаючи той факт, що всі запити до системи здійснювалися виключно текстом без коду.

Виведемо графік отриманої регресійної моделі для тестових даних та прогнозованої мінімальної ціни (див. рис. 7.18).

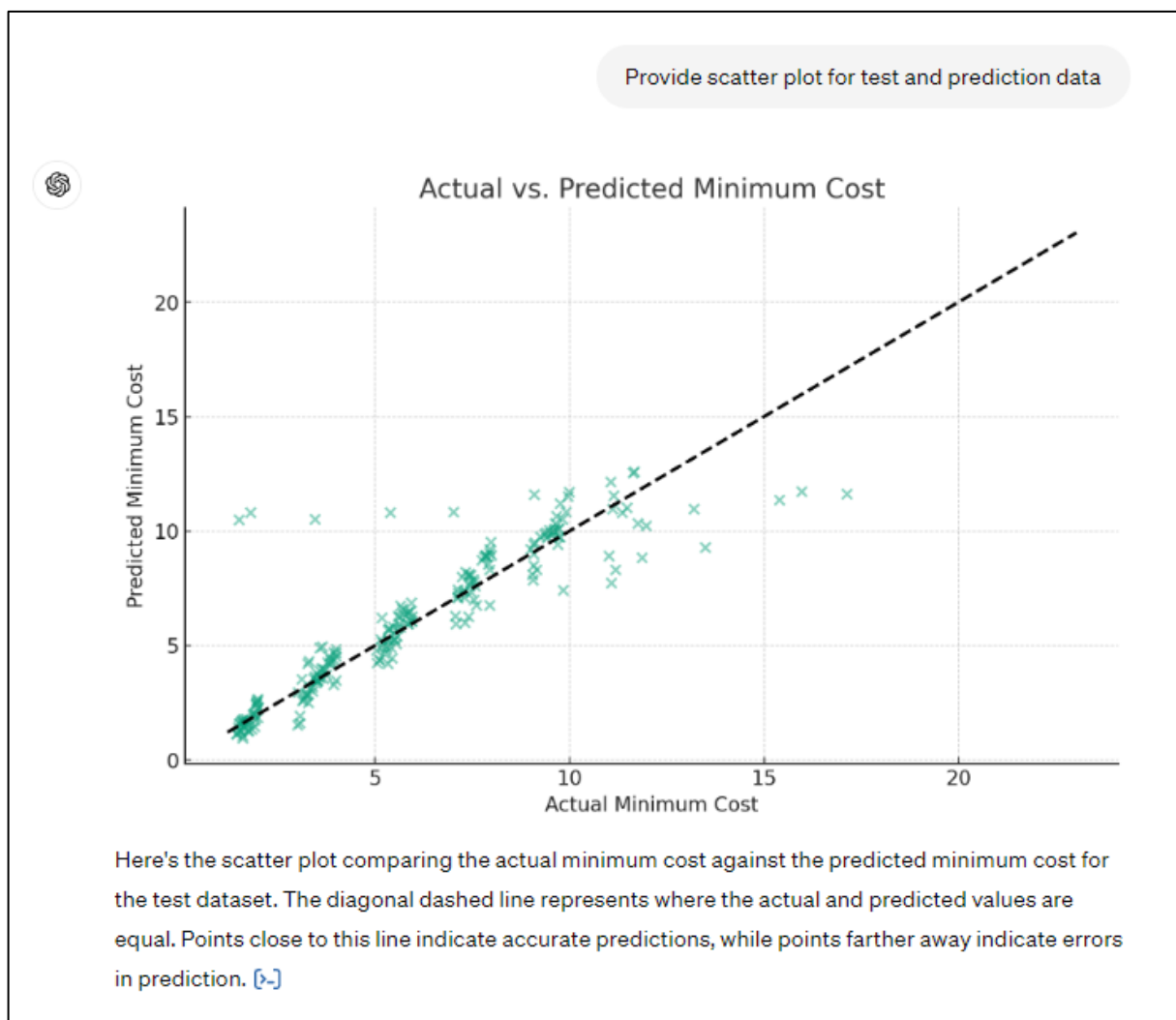


Рисунок 7.18 – Графік для отриманої регресійної моделі (виконано самостійно)

До графіку модель роз'яснення: «Ось діаграма розсіювання, що порівнює фактичну мінімальну вартість із прогнозованою мінімальною вартістю для тестового набору даних. Діагональна пунктирна лінія показує, де фактичні та прогнозовані значення збігаються. Точки, розташовані поблизу цієї лінії, вказують на точні прогнози, тоді як точки, розташовані далі, вказують на помилки в прогнозі.».

На графіку видно, що більшість точок знаходяться доволі близько до пунктирної лінії, що свідчить про точність отриманої регресійної моделі.

ВИСНОВКИ

У результаті виконання даної кваліфікаційної роботи були виконані поставлені задачі, а саме:

- зібрано та систематизовано теоретичний матеріал щодо транспортної логістики;
- описана технічна частина;
- приведені основні аспекти використання нейронних мереж у транспортній логістиці;
- сформовано та вирішено задачу багатокритеріального аналізу за допомогою симплекс методу;
- порівняно реалізації аналізу даних: програмну та зі застосуванням штучного інтелекту.

Кваліфікаційна робота описує основні переваги та недоліки застосування тих чи інших підходів та технологій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is logistics [Електронний ресурс] – Режим доступу до ресурсу: <https://quivo.co/us/what-is-logistics-simply-explained/>
2. Software component architecture in supply chain management [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/figure/Software-component-architecture-in-supply-chain-management_fig4_222833882.
3. GPT vs. BERT [Електронний ресурс] – <https://softteco.com/blog/bert-vs-chatgpt>.
4. OpenAI Pricing [Електронний ресурс] – Режим доступу до ресурсу: <https://openai.com/pricing>.
5. RoBERTa: An optimized method for pretraining self-supervised NLP systems [Електронний ресурс] – Режим доступу до ресурсу: <https://ai.meta.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/>.
6. YOLO: You Only Look Once [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/nerd-for-tech/yolo-you-only-look-once-65ea86104c51>.
7. Майбутнє кастомних AI-рішень [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ranktracker.com/uk/blog/the-future-of-custom-ai-solutions-predictions-from-industry-experts/>