НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ)

Інформаційних технологій

ПОГОДЖЕНО	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
Декан факультету (Директор ННІ)	**Завідувач кафедри**
Інформаційних технологій	Комп'ютерних наук
(назва факультету (ННІ))	(назва кафедри)
_____ Ігор Болбот	_____ Белла Голуб
(підпис) (ім'я ПРІЗВИЩЕ)	(підпис) (ім'я ПРІЗВИЩЕ)
"___" _____ 20_ р.	"___" _____ 20_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Система підтримки прийняття рішень керівництвом транспортної компанії в питаннях логістики

Спеціальність
122 «Комп'ютерні науки»
(код і найменування)

Освітня програма
«Інформаційні управляючі системи і технології»
(назва)

Орієнтація освітньої програми
освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми
к.т.н., доцент _____ Белла Голуб
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи
к.ф.-м.н., доцент _____ Яна Криворучко
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Виконав _____ Владислав Гринчук
(підпис) (ім'я ПРІЗВИЩЕ здобувача)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

доцент, к.т.н. _____ Голуб Б.Л.
(науковий ступінь, вчене звання) (підпис) (ПІБ)
" 10 " листопада 2024 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Гринчук Владислав Юрійович
(прізвище, ім'я, по батькові)

Спеціальність 122 "Комп'ютерні науки"
(код і найменування)

Освітня програма Інформаційні управляючі системи і технології
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи Система підтримки прийняття рішень керівництвом транспортної компанії у питаннях логістики

затверджена наказом проректора НУБіП України від " 1 " листопада 2024 р. № 1964 "С"

Термін подання завершеної роботи на кафедру 14 листопада 2025 р.
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи Набори даних відкритого доступу від логістичних компаній та геопросторові та офіційні державні дані.

Перелік питань, що підлягають дослідженню:

1. Аналіз проблем при прийнятті логістичних рішень транспортною компанією.
2. Визначення впливу зовнішніх та внутрішніх факторів на продуктивність при виконанні логістичних завдань.
3. Дослідження можливості застосування OLAP та Data Mining методів для підвищення ефективності виконання логістичних задач.
4. Дослідження результатів системи підтримки прийняття рішень керівництвом транспортної компанії у питаннях логістики

Перелік графічного матеріалу (за потреби)

Дата видачі завдання " 7 " листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Криворучко Я.С.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Гринчук В.Ю.

ЗМІСТ

1 ВСТУП

1.1 Актуальність теми

У сучасних умовах цифрової логістики щодня накопичуються великі обсяги різноманітних даних: інформація про маршрути, погодні умови, швидкість руху транспорту, рейтинги агентів, клієнтські характеристики тощо. Їхній ручний аналіз є малоефективним, а класичні статистичні підходи не дозволяють виявляти складні взаємозв'язки між факторами доставки.

Застосування методів інтелектуального аналізу даних (Data Mining) відкриває можливість автоматизовано виявляти закономірності, формувати прогнози та оцінювати ризики затримок. Однак використання окремих алгоритмів (наприклад, лише кластеризації чи класифікації) часто дає обмежене уявлення про логістичні процеси.

Тому доцільним є комбінування методів Data Mining — кластеризації, класифікації та пошуку асоціативних правил — що забезпечує більш глибокий аналітичний підхід, дозволяє сегментувати маршрути, навчати моделі для окремих кластерів та формулювати зрозумілі правила для управлінських рішень. Крім того, використання OLAP-технологій дає змогу здійснювати описовий аналіз логістичних показників (час доставки, довжина маршруту, середня ефективність), що доповнює результати інтелектуального аналізу та сприяє комплексному розумінню процесів у транспортній лоістиці.

1.2 Об'єкт і предмет дослідження

Об'єктом дослідження є процеси планування, організації та виконання доставок у транспортній логістиці.

Предметом дослідження є методи та інформаційні технології інтелектуального аналізу даних спрямовані на підвищення ефективності прийняття рішень у транспортній компанії шляхом комбінування алгоритмів кластеризації, класифікації та пошуку асоціативних правил а також застосування OLAP для описового аналізу та звітування.

1.3 Мета-дослідження

Метою роботи є дослідження та оцінка ефективності комбінованого підходу до інтелектуального аналізу логістичних даних, що поєднує методи кластеризації, класифікації та пошуку асоціативних правил із використанням OLAP-аналітики для виявлення взаємозв’язків між факторами, які впливають на строки доставки.

1.4 Завдання дослідження

Для досягнення поставленої мети в роботі необхідно було вирішити такі науково-дослідницькі завдання:

1. Провести аналіз сучасних методів інтелектуального аналізу даних, що застосовуються в транспортній логістиці, та визначити їх придатність для дослідження ефективності перевезень.
2. Підготувати логістичні дані до подальшої обробки: виконати очищення, нормалізацію та структурування інформації з бази даних, що містять параметри маршрутів, транспортних засобів, категорій доставок, погодних умов, трафіку, рейтингу агентів та час доставки.
3. Виконати кластеризацію методом K-Means для виявлення групи маршрутів з подібними характеристиками та виділення сегментів, між якими буде проводитися подальший аналіз.
4. Провести класифікаційне дослідження методом Наївного Баєса (Naive Bayes) для оцінки точності прогнозування взяття доставки між кожним кластером та визначення впливу окремих факторів на результат.
5. Застосовувати алгоритм пошуку асоціативних правил (Apriori) для виявлення закономірностей між такими атрибутами, як регіон (Area), тип транспорту (Vehicle), категорія замовлення (Category), погодні умови (Weather), стан трафіку (Traffic), частина доби (DayPart), рейтинг агента (AgentRating) та час виконання доставки (Delivery_Time).
6. Побудувати OLAP-куб для описового аналізу ключових показників маршрут, середньої швидкості - і сформувати звіти, для прийняття управлінських рішень.
7. Виконати порівняльний аналіз результатів кластеризації, класифікації та пошук асоціативних правил, узагальнити виявлені закономірності та зробити висновки щодо доцільності використання комбінованого підходу в системах підтримки та прийняття рішень транспортних компаній.

1.5 Методи дослідження

У роботі використано комплекс методів статистичного, аналітичного та інтелектуального аналізу даних, реалізованих у середовищах Microsoft SQL Server, Visual Studio та Python.

Метод кластеризації K-Means застосовано для поділу логістичних даних на групи (кластери) за спільними характеристиками маршрутів, погодних умов, транспортних засобів, рейтингу агентів та часу доставки. Це дало змогу виділити сегменти перевезень із подібними властивостями.

Метод класифікації Наївного Баєса (Naive Bayes) використовується для дослідження точності прогнозування затримок доставки в межах кожного кластера та оцінювання впливу ключових факторів на результати.

Метод пошуку асоціативних правил (Apriori) застосовано для виявлення стійких взаємозв'язків між такими атрибутами, як регіон (Area), тип транспорту (Vehicle), категорія замовлення (Category), погодні умови (Weather), стан трафіку (Traffic), частина доби (DayPart), рейтинг агента (AgentRating) та час виконання доставки (Delivery_Time).

OLAP-аналіз реалізовано в середовищі Microsoft Visual Studio з використанням SQL Server Analysis Services (SSAS). На основі підготовлених даних створено багатомірний OLAP-куб, у якому визначено виміри, ієрархію, ключові показники ефективності (KPI) та агрегати для подальшого аналізу.

У межах куба реалізовано обчислення основних показників: середнього часу доставки, середньої тривалості маршруту, співвідношення фактичного та запланованого часу, ефективності агентів. Для подальшої аналітики побудовано звіти у Microsoft SQL Server Reporting Services (SSRS) та візуалізації Visual Studio, які забезпечують динамічне представлення результатів дослідження.

Програмні обчислення в частині Data Mining виконано в середовищі Python з використанням бібліотек pandas, scikit-learn та mlxtend. Обмін даними відбувався через підключення до Microsoft SQL Server (представлення vw_MiningInput). Результати зберігалися у форматі CSV.

1.6 Наукова новизна

1. Досліджено ефективність комбінованого підходу до інтелектуального аналізу логістичних даних, який поєднує методи кластеризації, класифікації та пошук асоціативних правил з методом виявлення закономірностей у показниках доставки.
2. Встановлено, що використання кластеризації K-середніх для подальшого аналізу результатів класифікації дозволяє оцінити стабільність роботи моделі Наївного Баєса в межах різних груп даних і визначити, в яких умовах вона демонструє вищу або нижчу точність прогнозування.
3. Показано, що застосування асоціативних правил дає змогу підвищити інтерпретацію результатів методів Data Mining, оскільки дозволяє простежити логічні залежності між атрибутами, які впливають на тривалість доставки.
4. Систематизовано аналітичний підхід до дослідження логістичних даних, у межах якого поєднано два незалежні модулі аналітики - OLAP (для описового аналізу показників доставки) та Data Mining (для дослідження прихованих закономірностей).

1.7 Апробація результатів дослідження

1. ІНТЕЛЕКТУАЛЬНІ МЕТОДИ АНАЛІЗУ ДАНИХ ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ У ТРАНСПОРТНІЙ ЛОГІСТИЦІ Гринчук В.Ю., науковий керівник Криворучко Я.С. - XVI Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА» 28-29 жовтня 2025 року.
2. ВИКОРИСТАННЯ ТЕХНОЛОГІЙ DATA MINING ПРИ ПЛАНУВАННІ ЛОГІСТИЧНИХ ОПЕРАЦІЙ ЗА ДОПОМОГОЮ ІНСТРУМЕНТІВ БІЗНЕС АНАЛІТИКИ Гринчук В.Ю., науковий керівник Криворучко Я.С.

- IX Міжнародна студентська наукова конференція «Актуальні питання та перспективи проведення наукових досліджень» 30.05.2025.

1.8 Структура магістерської роботи

Робота має обсяг у 74 сторінки, 16 використаних джерел, 3 додатки, та 7 розділів. Перший розділ – Вступ, в ньому наведено основні відомості по дослідженню. Другий розділ – Аналіз предметної області, в ньому проведено аналіз області транспортної логістики. Третій розділ – моделювання системи, в ньому створена концептуальна модель системи, та побудовані допомічні діаграми. Четвертий розділ – Розробка системи, в ньому представлена розробка модулів та елементів системи, наведені методологія та застосування. П’ятий розділ – Результати дослідження, в якому показано остаточні результати роботи моделей, порівняння методів, та розробки системи. Шостий розділ – Висновки, та Сьомий – Список використаних джерел.

2 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Аналіз предметної області транспортної логістики

Транспортна логістика є ключовим компонентом у системі постачання товарів, що забезпечує своєчасне та ефективне переміщення вантажів між учасниками ланцюга постачання. У сучасних умовах цифровізації транспортні компанії оперують великими масивами даних, які охоплюють маршрути, погодні умови, стан дорожнього руху, тип транспортного засобу, час доби, характеристики клієнтів і результати доставок.

Актуальність аналітичного підходу до управління логістикою полягатиме в тому, що ручний аналіз таких даних є неефективним через обсяг і швидкість їх накопичення. Для підвищення якості прийняття управлінських рішень необхідно застосовувати методи багатомірного аналізу та дата майнінгу, здатні автоматично виявляти закономірності у виконанні перевезень.

У межах цього дослідження, як емпіричну базу, використано Amazon Delivery Dataset (платформа Kaggle), що містить інформацію про 43 тисячі виконаних доставок із зазначенням географічних, погодних, транспортних та часових характеристик. Цей набір даних є репрезентативним прикладом для дослідження ефективності логістичних процесів та формування моделей прийняття управлінських рішень.

2.2 Рівень розвитку інформаційних систем у транспортній логістиці

Сучасні транспортні компанії активно впроваджують інформаційні системи управління перевезеннями (TMS — Transport Management System), які забезпечують облік, моніторинг і планування маршрутів. Однак більша частина таких систем обмежується операційними функціями: зберіганням даних про доставку, відстеженням транспортних засобів і фіксацією часу виконання замовлень.

Традиційні бази даних (OLTP-системи) призначені для транзакційних операцій, але не дозволяють здійснювати глибокий аналітичний чи прогностичний аналіз. Для цього використовуються OLAP-сховища, що підтримують багатомірну агрегацію даних, та технології Data Mining, які виявляють приховані закономірності, групи або правила у великих наборах даних.

У межах проекту реалізовано аналітичну платформу, яка поєднує підходи OLAP та Data Mining у середовищі Microsoft SQL Server.

На рівні OLAP створено багатовимірне сховище з фактами доставок і вимірами (**DimWeather, DimDate, DimVehicle, DimTraffic, DimArea, DimCategory, DimAgent, DimTime, FactDeliveries**).

На рівні DataMining розроблено комбінації методів кластеризації, класифікації та пошуку асоціативних правил які забезпечують можливість

дослідження структури логістичних даних із різних аналітичних ракурсів. Кластеризація дозволяє виокремити групи маршрутів за подібними характеристиками, класифікація методом Наївного Байєса дає змогу оцінити точність прогнозування затримок у межах кожного кластера, а асоціативні правила дозволяють встановити зв'язки між атрибутами, що впливають на тривалість доставки. Такий підхід дає змогу не лише аналізувати розподіл ефективності моделі за кластерами, а й глибше зрозуміти логіку взаємодії факторів транспортного процесу.

Отримані результати стали основою для формування вдосконаленої аналітичної системи, в якій OLAP-модуль забезпечує багатовимірний описовий аналіз показників доставки, а Data Mining-модуль - дослідницьку обробку даних із використанням комбінованих методів. Разом вони формують єдине інформаційне середовище, що дозволяє проводити як традиційний звітний аналіз, так і поглиблене дослідження закономірностей у транспортній логістиці.

2.3 Проблематика прийняття управлінських рішень у сучасних транспортних компаніях

Незважаючи на зростання кількості IT-рішень у сфері логістики, проблема прийняття управлінських рішень залишається складною через такі чинники:

- висока варіативність зовнішніх факторів, зокрема погодних умов і дорожнього трафіку;
- нерівномірність навантаження маршрутів, що призводить до затримок і перевитрат ресурсів;
- брак інтеграції аналітичних даних між операційними та стратегічними рівнями управління;
- обмежена аналітична інтерпретація даних, коли результати фіксуються, але не використовуються для прогнозування чи планування.

Для подолання цих недоліків необхідні системи, що не просто зберігають інформацію про перевезення, а дозволяють керівництву виявляти закономірності та приймати рішення на основі аналітичних висновків. Те саме стосується дослідження поєднання OLAP та Data Mining у контексті транспортної логістики, а також комбінації методів Data Mining.

2.4 Постановка задачі дослідження

Сучасна транспортна логістика характеризується високою динамікою процесів і великою кількістю факторів, що впливають на якість доставки. Управлінські рішення в транспортній компанії грунтуються на аналізі показників – часу виконання замовлення, маршруту, погодних умов, завантаженості доріг, типу транспортного засобу, характеристик клієнта та виконавця. При цьому значна частина даних має приховані взаємозв'язки,

виявлення яких є складним завданням для традиційних методів статистичного аналізу.

У зв'язку з цим виникає наукова проблема — забезпечення можливості багатовимірного та інтелектуального аналізу логістичних даних для підвищення обгрунтованості управлінських рішень. Її вирішення потребує використання двох взаємодоповнювальних підходів: OLAP-аналітики (для структурованого описового аналізу даних) та Data Mining (для автоматичного виявлення закономірностей і дослідження факторів, що впливають на строки доставки).

Для досягнення мети передбачається:

- Провести системний аналіз логістичних даних транспортної компанії та будувати багатовимірну модель сховища з основними вимірами та фактами доставки;
- Виконати аналітичні дослідження методами кластеризації (K-Means), класифікації (Наївний Байєс) та пошуку асоціативних правил (Apriori) з використанням середовища Python та SQL Server;
- Оцінити ефективність комбінованого підходу Data Mining шляхом порівняння точності класифікації та стабільності результатів у межах кластерів;
- Провести описовий аналіз показників доставки в OLAP-кубі та сформувати звіти для узагальнення результатів дослідження.

3 МОДЕЛЮВАННЯ СИСТЕМИ

3.1 Концептуальна модель системи підтримки прийняття рішень

Концептуальна модель системи відображає логіку побудови аналітичного середовища, що поєднує методи OLAP-аналітики та інтелектуального аналізу даних (Data Mining) для підтримки управлінських рішень у транспортній логістиці. Основна ідея полягає у поєднанні декількох методів Data Mining у межах єдиного дослідницького процесу, що забезпечує більш глибоке розуміння закономірностей у логістичних даних та підвищує пояснювальність моделей.

Система охоплює три рівні:

- Операційний рівень (OLTP): забезпечує збір і зберігання первинних даних про процеси доставки: замовлення, маршрути, погодні умови, транспорт, рейтинг агентів, час виконання, трафік. Ці дані зберігаються в операційній базі транспортної компанії і виступають джерелом для подальшого аналізу.
- Аналітичний рівень (OLAP): реалізований у середовищі Microsoft SQL Server Analysis Services (SSAS). На цьому рівні побудовано багатовимірне сховище даних, структуроване за вимірами DimVehicle, DimClient, DimWeather, DimTraffic, DimTime, DimArea, DimCategory, DimDate, DimAgent та таблицею фактів FactDeliveries. OLAP-рівень забезпечує можливість формування звітів, обчислення KPI та виконання порівняльного аналізу показників доставки за різними зрізами (час, відстань, середня швидкість, ефективність агентів).
- Інтелектуальний рівень (Data Mining): є основним об'єктом дослідження. Тут реалізовано комбінацію трьох методів — кластеризації (K-Means), класифікації (Наївний Байєс) та пошуку асоціативних правил (Apriori). Кластеризація використовується для виділення груп маршрутів із подібними характеристиками, класифікація оцінює точність прогнозів у межах кожного кластера; асоціативні правила формують логічні пояснення отриманих результатів, визначаючи, які фактори впливають на строки доставки. Такий підхід забезпечує взаємодоповнення методів, коли результати одного етапу (кластеризації) уточнюють умови для іншого (класифікації), а асоціативні правила підвищують інтерпретованість отриманих моделей

Таким чином, концептуальна модель системи поєднує описову аналітику (OLAP) та пояснювальний інтелектуальний аналіз (Data Mining). Це дає змогу проводити не лише моніторинг основних показників доставки, а й виявляти приховані закономірності та пояснювати їх вплив на ефективність логістичних процесів.

3.2 Функціональне моделювання системи

Для опису функцій системи використано **UML-нотацію.** Основним користувачем є керівництво транспортної компанії, яке виступає гіпотетичним суб'єктом прийняття управлінських рішень.

Діаграма прецедентів. Діаграма прецедентів(рис. 1.1) відображає основні ролі користувачів у системі підтримки прийняття рішень керівництвом транспортної компанії та їхні взаємодії з функціональними модулями. Адміністратор системи відповідає за оновлення даних сховища, налаштування OLAP-куба та виконання операцій Data Mining. Аналітик формує аналітичну звітність і визначає ключові показники ефективності (KPI). Керівник компанії, у свою чергу, взаємодіє з аналітичними звітами та приймає управлінські рішення на їх основі. Такий підхід відображає логіку використання системи різними ролями в процесі управління логістичними операціями.

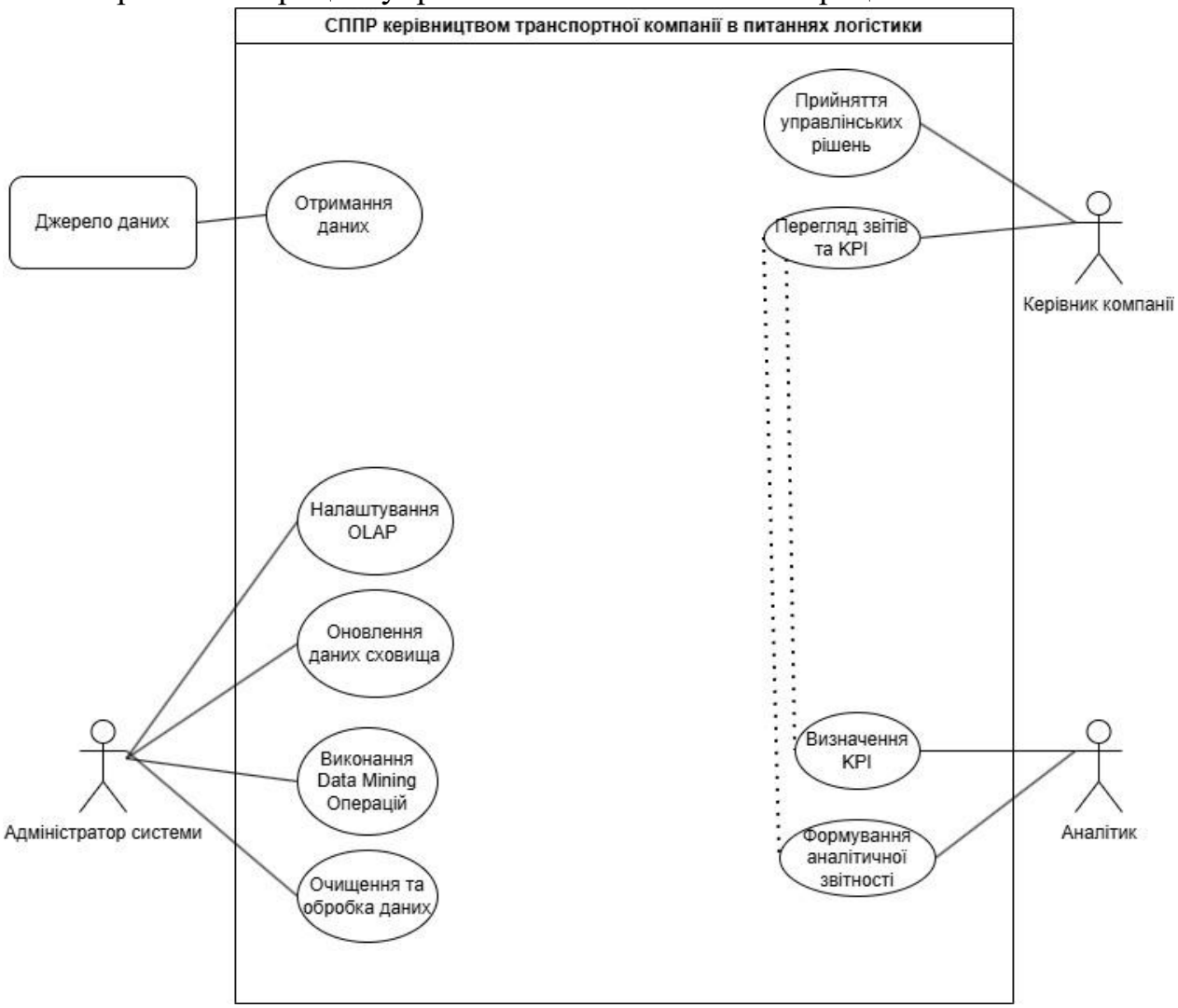

Рис. 1.1 Діаграма прецедентів

Діаграма активності. Діаграма активності(рис. 1.2) описує послідовність дій у межах функціонування системи - від завантаження даних до прийняття управлінських рішень. Процес починається з дій адміністратора, який ініціює завантаження даних та запуск ETL-процесу. Після формування агрегатів у

сховищі даних здійснюється обробка куба OLAP, а далі виконання операцій Data Mining (кластеризація K-Means, класифікація Naïve Bayes, пошук правил Apriori). Отримані результати проходять через пояснювальний шар, де моделі взаємопов'язуються між собою для підвищення інтерпретованості. Завершується процес етапом звітування, на якому керівник аналізує KPI, результати аналітики та приймає рішення.

Рис. 1.2 Діаграма активності

Топологія системи. Діаграма топології(рис. 1.3) відображає архітектурну структуру системи, розподіл функцій між компонентами та канали обміну

даними. Центральним елементом є системний сервер із модулем збору даних, який інтегрується з зовнішніми джерелами - службами виконання доставок, GPS-трекерами, метеорологічною службою, дорожньою інформацією та системою управління складом. На сервері бази даних формується операційне середовище, з якого дані завантажуються у сховище (сервер СД) для подальшого аналітичного опрацювання. Робочі станції аналітика та керівника забезпечують взаємодію користувачів із системою через модулі Data Mining, OLAP та звітності, реалізуючи замкнений цикл отримання, аналізу та візуалізації логістичних даних.

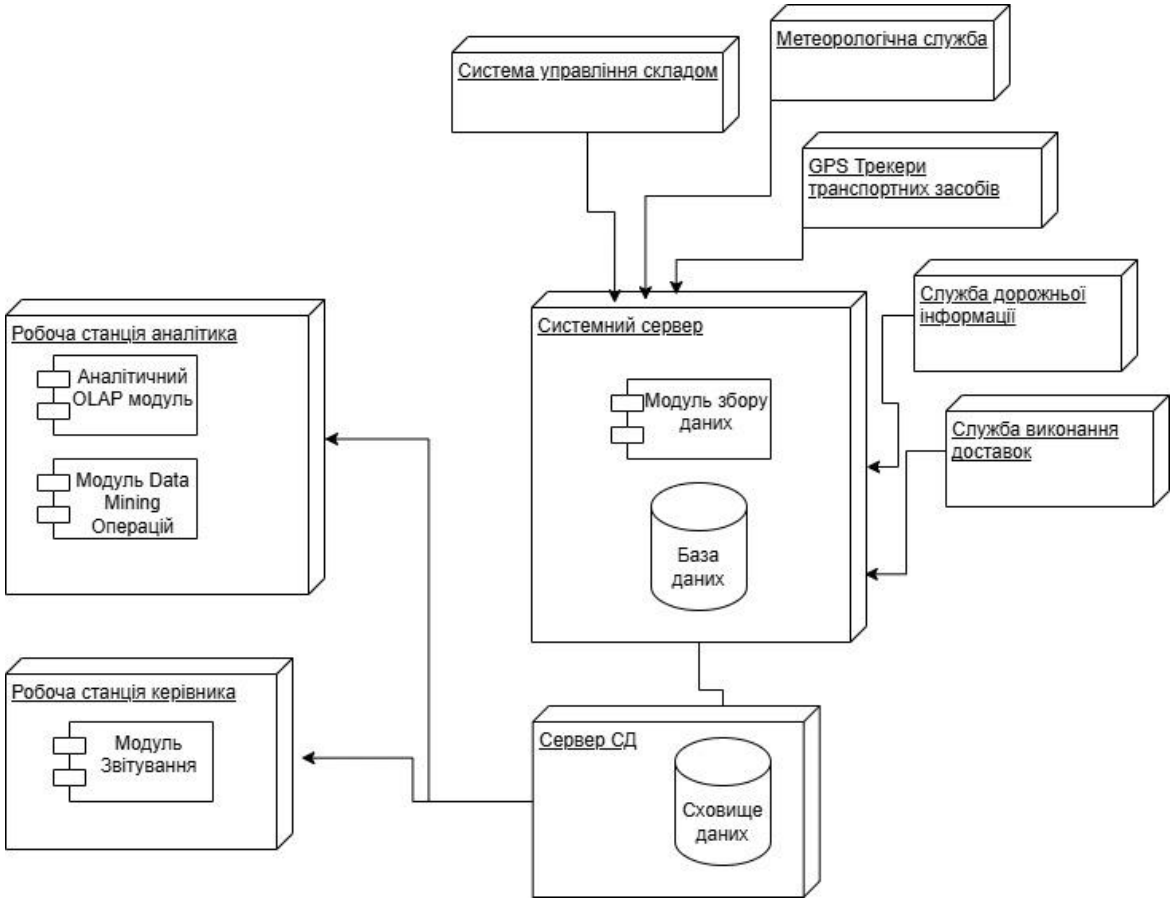

Рис. 1.3 Топологія системи

3.3 Висновки до розділу

У цьому розділі проведено моделювання системи підтримки прийняття рішень для транспортної компанії.

Було використано комбінований підхід — функціональне та об'єктно-орієнтоване моделювання, що дозволило описати як зовнішню поведінку системи (через прецеденти, послідовності, активності), так і її внутрішню структуру (класи, зв'язки, компоненти).

Отримані моделі є основою для реалізації системи в наступному розділі, де розглядаються технічна реалізація, структура збереження даних, аналітичні процедури та комбінації методів Data Mining.

4 РОЗРОБКА СИСТЕМИ

4.1 Середовище та засоби реалізації

Реалізацію системи підтримки прийняття рішень для транспортної компанії виконано у середовищі Microsoft SQL Server Management Studio (SSMS) та Visual Studio 2022 із використанням модулів SQL Server Integration Services (SSIS) і SQL Server Analysis Services (SSAS). Для реалізації алгоритмів інтелектуального аналізу даних застосовано мову Python 3.13.3 із бібліотеками pandas, scikit-learn, mlxtend, numpy, pyodbc, matplotlib.

Така комбінація забезпечила створення комплексної аналітичної системи, у межах якої OLAP модуль використовується для описового аналізу показників доставки, а Data Mining модуль – для дослідження прихованих закономірностей у даних[1]. Реалізоване поєднання методів кластеризації, класифікації та пошуку асоціативних правил дозволяє не лише оцінювати точність прогнозів, але й пояснювати їх через взаємозв'язки між факторами, що впливають на ефективність логістичних процесів[2].

4.2 Створення сховища даних

Джерело даних

Початковим джерелом виступає Amazon Delivery Dataset, з відкритого доступу на Kaggle[3], він містить інформацію про:

- Ідентифікатор замовлення (Order_ID),
- Час доставки (Delivery Time),
- Координати початкової та кінцевої точок доставки (Store_Latitude/Longitude, Drop_Latitude/Longitude)
- Погодні умови (Weather),
- Рівень трафіку (Traffic),
- Тип транспортного засобу (Vehicle),
- Регіон доставки (Area),
- Рейтинг виконавця (Agent_Rating),
- Вік виконавця (Agent_Age),
- Категорію замовлення (Category),
- Дату доставки (Order_Date).

4.3 Структура бази даних

На етапі створення сховища реалізовано зіркову схему (Star Schema) з центральною фактною таблицею FactDeliveries, та 8 таблицями вимірами. FactDeliveries – Фактичні показники доставки, час, відстань, та зовнішні ключі вимірів.

Для визначення відстані між точками відправлення та доставки в таблиці FactDeliveries(рис. 2.1) була використана просторова функція SQL

Geography. Вона дозволяє обчислювати геодезичну відстань між координатами (широта та довжина) за допомогою методу STDistance(), який враховує кривизну поверхні Землі[4]. Таким чином, значення атрибута Distance_Km формується одночасно під час завантаження даних у сховище, забезпечуючи високу точність розрахунків маршруту.

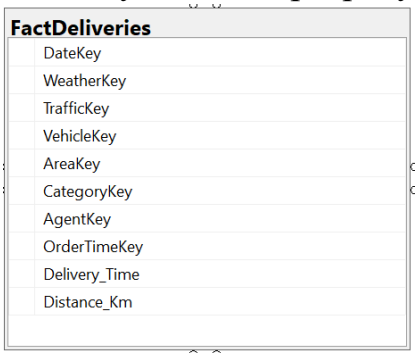

Рис. 2.1 Таблиця фактів

DimDate(рис. 2.2) – Календарна інформація (повна дата, квартал, місяць, день)

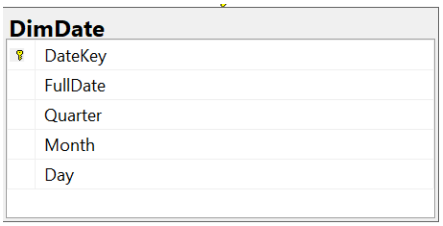

Рис. 2.2 Вимір дати

DimTime(рис. 2.3) – Час доставки (година, хвилина, час доби)

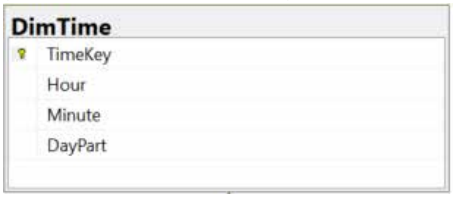

Рис. 2.3 Вимір часу

DimArea(рис. 2.4) – Регіон доставки

Рис. 2.4 Вимір регіонів

DimWeather(рис. 2.5) – Погодні умови

Рис. 2.5 Вимір погоди

DimTraffic(рис. 2.6) – Рівень завантаженості доріг

Рис. 2.6 Вимір трафіку

DimAgent(рис. 2.7) – Дані виконавця, його рейтинг та вік

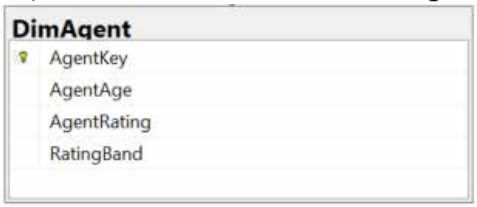

Рис. 2.7 Вимір агентів

DimCategory(рис. 2.8) – Категорія товарів доставки

Рис. 2.8 Вимір категорій

DimVehicle(рис. 2.9) – Тип транспортного засобу виконавця

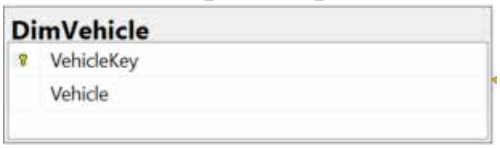

Рис. 2.9 Вимір транспорту

Схему реалізовано у базі Amazon Delivery(рис. 2.10), створеній у SSMS[5].

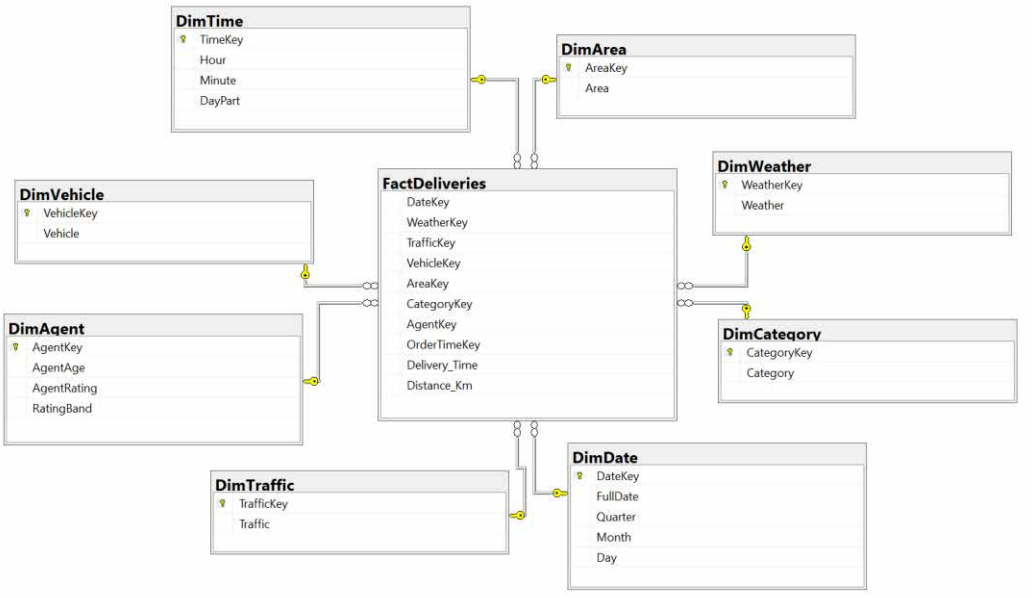

Рис. 2.10 Схема сховища даних

Процес завантаження даних (ETL-процес) у сховище Amazon Delivery реалізовано засобами SSIS Data Flow. На початковому етапі з CSV-файлів Amazon Delivery Dataset виконується імпорт логістичної інформації. Дані проходять проміжну обробку очищення від пропусків, приведення до потрібних типів і фільтрацію аномалій. Далі здійснюється перетворення, що включає

створення ключів вимірів, нормалізацію та агрегування показників. Завершальним кроком є завантаження очищених і структурованих даних у таблиці Dim* та FactDeliveries бази даних Amazon Delivery, які надалі використовуються для побудови OLAP-куба та виконання аналітичних запитів.

4.4 Побудова OLAP-куба

Структура багатовимірної моделі. Побудова кубу здійснюється шляхом створення Data Source View(рис. 2.11), у якому відображено всі зв'язки між таблицями фактів і вимірів. Для кожного виміру визначаються ієрархії, що дозволяють деталізувати показники у звітах: наприклад, у вимірі DimDate ієрархія має рівні Рік → Квартал → Місяць → День, а у вимірі DimTime — Частина доби → Година. Така ієрархічна структура забезпечує можливість аналізу показників доставки на різних рівнях деталізації, що є основою для подальшої побудови управлінських звітів. На основі цієї структури створено куб DeliveriesCube, у якому визначено дві головні міри Delivery_Time Та Distance_Km. Додатково передбачено можливість побудови похідних показників для оцінки ефективності доставки, таких як середній час виконання, середня довжина маршруту та коефіцієнт відхилення фактичного часу від нормативного. Завдяки цим показникам забезпечується аналітичне відображення ключових аспектів логістичної діяльності компанії. У результаті побудований куб забезпечує швидкий доступ до агрегованих показників, формування багатовимірних зрізів і створення звітів у середовищі SQL Server Reporting Services (SSRS). Це дозволяє керівництву компанії виконувати динамічний моніторинг ефективності доставки, порівнювати показники між різними регіонами та оперативно реагувати на зміни у логістичних процесах.

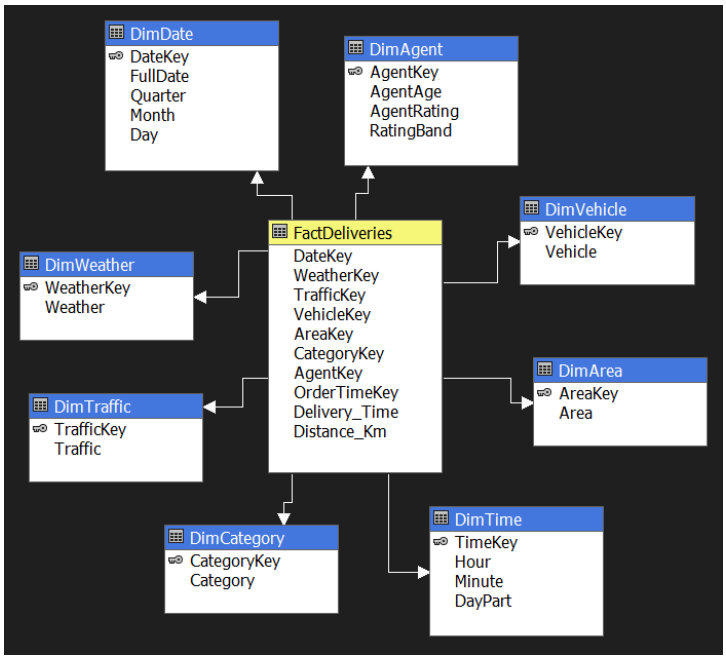

Рис. 2.11 OLAP куб

Розробка кубу. Розроблення багатовимірного кубу виконано у середовищі Microsoft Visual Studio 2022 із використанням аналітичного модуля SQL Server Analysis Services (SSAS). На основі раніше створеного сховища даних побудовано модель типу зірка (Star Schema), у якій центральне місце займає таблиця фактів FactDeliveries, пов'язана з вимірами через зовнішні ключі. Така структура забезпечує логічну простоту, високу швидкість обробки запитів та зручність у подальшому проєктуванні кубу.

На початковому етапі створюється Data Source - підключення до бази даних, розміщеної у середовищі SQL Server Management Studio (SSMS). Після цього формується Data Source View, у якому відображено всі таблиці фактів і вимірів, а також визначено зв'язки між ними. У цій моделі таблиця FactDeliveries містить основні числові показники логістичних процесів - час виконання доставки (Delivery_Time) та довжину маршруту (Distance_Km). Кожен запис у таблиці фактів має посилання на відповідні ключі вимірів: дати (DateKey), часу (OrderTimeKey), району (AreaKey), типу транспорту (Vehiclekey), агента (AgentKey), категорії замовлення (CategoryKey), а також погодних і дорожніх умов (WeatherKey, Traffickey). На основі цих зв'язків створюються окремі виміри (Dimensions), що описують різні аспекти логістичних операцій.

Вимір DimDate містить поля FullDate, Quarter, Month, Day і дозволяє виконувати аналітичні запити у часовому розрізі. Вимір DimTime деталізує процес за атрибутами Hour, Minute Та DayPart, що дає змогу аналізувати залежність ефективності доставки від частини доби. Вимір DimAgent зберігає характеристики виконавців — вік, рейтинг і рейтинг-банд (RatingBand), що є важливою змінною для оцінки якості роботи персоналу. Окремі виміри DimWeather і DimTraffic містять інформацію про погодні та дорожні умови, які безпосередньо впливають на час доставки. Додатково створено виміри DimArea,

DimVehicle та DimCategory, які відповідають за просторові, технічні та класифікаційні компоненти процесу перевезень.

Після визначення всіх вимірів формується структура кубу DeliveriesCube, у якій задано головні міри Delivery_Time Та Distance_Km. У процесі проєктування було додано обчислювані атрибути, що використовуються для подальшого формування KPI, зокрема Average Delivery Time (середній час доставки) та Average Distance per Delivery (середня довжина маршруту). Ці показники створюються безпосередньо у середовищі Visual Studio засобами MDX-виразів і автоматично оновлюються при кожному оновленні кубу.

Для забезпечення багатовимірного аналізу у вимірах визначено ієрархії, які дозволяють користувачам виконувати деталізацію та агрегацію даних на різних рівнях. Наприклад, у вимірі DimDate ієрархія побудована за рівнями Рік → Квартал Місяць День, у вимірі DimAgent – RatingBand → AgentRating → AgentID, а у вимірі DimArea — Область Район. Таке ієрархічне структурування забезпечує можливість аналітичного дослідження факторів на різних рівнях узагальнення та підвищує ефективність візуалізації у звітах.

Завершальним етапом проєктування є розгортання кубу на сервері SSAS і виконання обробки (processing) даних(рис. 2.12). У процесі обробки здійснюється завантаження агрегованих значень, після чого куб стає доступним для аналітичних запитів у середовищі SQL Server Management Studio, Excel або Power BI. Це дозволяє використовувати куб як основне джерело для побудови інтерактивних звітів і візуалізації показників у рамках системи підтримки прийняття рішень.

```
------ Deploy started: Project: Amazon_delivery, Configuration: Development ------
Performing an incremental deployment of the 'Amazon_delivery' database to the 'WootDaFlip-PC' server.
No changes detected. The Amazon_delivery database on the WootDaFlip-PC server is up-to-date.
Deploy complete -- 0 errors, 0 warnings
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
========== Build completed at 0:39 and took 07,288 seconds ==========
========== Deploy: 1 succeeded, 0 failed, 0 skipped ==========
========== Deploy completed at 0:39 and took 07,288 seconds ==========
```

Рис. 2.12 Деплоймент кубу

4.5 Оцінка ефективності за KPI та розробка звітів

У межах розробки системи підтримки прийняття рішень було створено набір ключових показників ефективності (Key Performance Indicators — KPI), що відображають якість та стабільність процесів доставки. Їхня реалізація здійснювалась у середовищі Microsoft Visual Studio 2022 на базі SQL Server Analysis Services (SSAS). Ці показники стали основою для побудови аналітичних звітів у SQL Server Reporting Services (SSRS), які забезпечують візуальний моніторинг логістичних параметрів.

Створення мір у кубі. На першому етапі було сформовано систему базових мір (Measures), які агрегують ключові числові показники з таблиці фактів FactDeliveries. До основних мір належать:

- Deliveries Count - загальна кількість доставок;
- Delivery Time - середній час виконання доставки;
- Distance Km середня довжина маршруту;
- OnTimeRate частка доставок, виконаних вчасно (менше 90 хвилин);
- Delay120Ratio — відсоток доставок із затримкою понад 120 хвилин;
- AvgDeliveryTime, AvgDistance, Agents Distinct — обчислювані показники для середніх значень часу, відстані та кількості виконавців.

Для підвищення точності аналізу також реалізовано допоміжні фракційні міри (OnTimeRate_Fraction, Delay120Ratio_Fraction), які використовуються у KPI для коректного розрахунку часткових показників у відсотках.

Обчислення проводились засобами MDX, що дозволяє виконувати динамічні агрегації по ієрархіях вимірів (дата, агент, регіон, категорія, трафік, погода, час.). Це дало змогу створити гнучку систему аналітичних залежностей, де кожен показник автоматично перераховується при фільтрації даних у звітах.

Формування KPI. На основі побудованих мір у середовищі SSAS створено п'ять ключових KPI, які відображають основні параметри ефективності доставки:

1. KPI Avg Delivery Time — середній час доставки у хвилинах;
2. KPI On-Time Rate — відсоток доставок, виконаних вчасно;
3. KPI Delay120 Ratio частка доставок з виконанням понад 120 хвилин;
4. KPI Speed Index — співвідношення відстані до часу доставки (характеризує швидкість руху);
5. KPI DES (Delivery Efficiency Score) — інтегральний показник, що відображає ефективність логістичних процесів.

Для кожного KPI задано мету (Goal), поточне значення (Value), статус і тренд. Ці параметри реалізовано у вигляді графічних індикаторів - шкал типу gauge із кольоровими зонами (зелена норма, жовта — середній рівень, червона — відхилення). Завдяки цьому забезпечується швидка візуальна оцінка стану логістичної системи без необхідності глибокого аналітичного аналізу[6].

KPI використовують міри, розраховані у кубі, а їхні цільові значення (Goal) визначено на основі статистичних середніх за історичний період. Наприклад, мета для середнього часу доставки - 90 хвилин, а для відсотка вчасних доставок — 70%. Усі KPI було згруповано у категорію Deliveries Performance, що дозволяє відстежувати їх у комплексі.

Побудова звітів. Для представлення результатів у зручній формі були розроблені кілька аналітичних SSRS-звітів, які використовують куб як джерело даних[7].

До основних звітів належать:

1. Порівняльний звіт за поточний та попередній періоди (28 днів) містить таблиці та діаграми для аналізу показників Avg Delivery Time, Deliveries Count, Delay 120Ratio за категоріями рейтингу агентів (Rating Band).

- Використано лінійні графіки середнього часу доставки та стовпчикові діаграми кількості доставок.
- Забезпечено можливість одночасного порівняння поточного й попереднього періодів, що дозволяє виявляти тенденції у динаміці доставки.

2. Звіт ефективності доставки по датах включає показники On Time Rate, Delay 120Ratio, Avg Delivery Time, відображені у часовому розрізі.

- Графік "Середній час доставки по датах" демонструє зміну ефективності протягом місяця.
- Графік "Частка вчасних доставок" дозволяє контролювати динаміку пунктуальності агентів.

3. Підсумковий KPI Dashboard інтегрований звіт із візуальними індикаторами, які показують поточне значення, ціль, статус і тренд для кожного KPI (Avg Delivery Time, Delay120 Ratio, DES, On-Time Rate, Speed Index).

- Кожен індикатор має кольорову шкалу, що дозволяє швидко оцінити, які з показників відповідають заданим нормам, а які потребують уваги.
- Розділ "Trend" відображає напрям зміни показника (покращення, стабільність або погіршення).

Побудова, використання, та комбінування моделей Data Mining

Для зручності завантаження необхідні дані з бази представлені у вигляді таблиці dbo.vw_MiningInput(рис. 2.13).

Рис 2.13 Представлення даних сховища

4.6 Класифікація (Наївний Байес)

Мета методу. Побудувати багатокласову модель, що дозволяє класифікувати час доставки замовлення у відповідні часові діапазони (0-60, 61-120, 120-180, 180+ хвилин) на основі історичних даних про доставку. Такий підхід дає змогу оцінити імовірність належності конкретної доставки до певного класу затримки, використовуючи як числові, так і категоріальні ознаки. Метод базується на теоремі Байеса з припущенням незалежності ознак, що забезпечує простоту обчислень і високу швидкість навчання навіть на великих вибірках.

Завантаження даних. Дані зчитуються з подання dbo.vw_MiningInput (SQL Server) через драйвер ODBC Driver 17 for SQL Server з використанням бібліотеки pyodbc.

Інженерія ознак. У процесі підготовки даних розраховується новий показник Pace_MinPerKm, який визначає середній темп доставки в хвилинах на кілометр: Pace_MinPerKm = Delivery_Time / Distance_Km**.** Це дозволяє врахувати не лише абсолютну тривалість доставки, а й ефективність руху на різних відстанях,

Визначення цільової змінної. На основі фактичного часу доставки формується категоріальна змінна delay_band, яка поділяє записи на інтервали:

- 0-60 – швидкі доставки,
- 61-120 – середні за тривалістю,
- 120-180 – довгі,
- 180+ - дуже довгі.

Випадки з відсутніми значеннями виключаються з подальшої вибірки.

Попередня обробка даних. Для числових ознак(Distance_Km, Delivery_Time, Pfce_MinPerKm, AgentRating) застосовується імп’ютеризація за медіаною та стандартизація (StandartScaler).

Для категоріальних ознак (Area, Vehicle, Category, Weather, Traffic, DayPart) використовується імп’ютеризація за найчастішим значенням і кодування методом One-Hot Encoding. Ці перетворення об’єднані у єдиний конвеєр ColumnTransformer.

Формування моделі. Створюється послідовний конвеєр (Pipeline), який поєднує етапи попередньої обробки (pre) і модель GaussianNB. Вибір моделі грунтується на тому, що розподіл числових ознак наближений до нормального, що відповідає передумовам Наївного Байеса[8].

Навчання та тестування. Дані поділяються у пропорції 80/20 (test_train_split) із забезпеченням стратифікації за класами, щоб зберегти пропорції міток у тренувальній і тестовій вибірках. Після навчання модель робить передбачення (predict), а резульиаим оцінюються за метриками точності (accuracy), F1-мірою та звітом класифікації (classification_report)[9].

Збереження результатів. Отримані показники зберігаються в csv файлі, у каталозі results, для зручності подальших комбінованих операцій.

Підсумок. Метод Наївного Байєса забезпечує надійну основу для первинного моделювання процесу доставки, поєднуючи простоту реалізації з імовірною природою оцінювання. Його використання дозволяє автоматизувати класифікацію замовлень за рівнем затримки та підготовлювати базову модель, яка служить еталоном для подальшого порівняння з іншими алгоритмами Data Mining.

4.7 Кластеризація методом K-Means

Метою застосування алгоритму K-Means є виявлення груп (кластерів) доставок із подібними характеристиками часу виконання, відстані, темпу виконання, та рейтингу виконавця. Такий поділ дозволяє виявити закономірності у продуктивності маршрутів, визначити типові моделі поведінки кур'єрів і сформувати аналітичну основу для оцінки ефективності логістичних процесів. Метод реалізовано мовою Python з використанням бібліотек pandas, numpy, scikit-learn, підключенням до SQL Server через ODBC та збереженням результатів у csv для подальшого комбінування.

Джерело даних і попередня обробка. Початкові дані завантажуються з представлення dbo.vw_MiningInput у базі Amazon Delivery. З цього джерела відбираються числові поля:

- Distance_Km – довжина маршруту;
- Delivery_Time – фактична тривалість доставки;
- AgentRating – оцінка виконавця клієнтом або системою.

На основі цих показників обчислюється похідна змінна:

$$Pace\_MinPerKm = \frac{Delivery\_Time}{Distance\_Km}$$

Яка характеризує середній темп доставки (у хвилинах на кілометр). Це дозволяє зіставляти ефективність маршрутів різної довжини.

Відсікання аномалій та нормалізація. Для забезпечення коректності аналізу застосовано два рівні очищення:

1. Базові санітарні фільтри, що виключають явні помилки вимірювань:
   - Відстан у межах (0.05; 200) км,
   - Час доставки в межах (3; 1000) хв,
   - Темп у межах (2; 60) хв/км.
2. **IQR-тримінг (Interquartile Range)** – метод усунення статистичних викидів. Для кожної ознаки обчислюються квартилі Q1 і Q3, після чого залишаються лише рядки, значення яких потравляють у діапазон:

$$[Q1 - 1.5 \times IQR; Q3 + 1.5 \times IQR]$$

Це дозволяє зберегти лише типові спостереження без екстремальних значень, що спотворюють центроїди кластерів. Після очищення дані масштабуються за допомогою RobustScaler, який виконує нормалізацію відносно медіани та міжквартильного розмаху. Такий підхід робить модель стійкою до решти потенційних викидів і забезпечує рівномірний внесок кожної ознаки у розрахунок відстаней.

Вибір кількості кластерів. Для визначення оптимальної кількості кластерів було використано два підходи метод силуетів (Silhouette Method) та метод ліктя (Elbow Method)[10]. Метод Silhouette оцінює якість кластеризації, порівнюючи відстань між об'єктами в межах одного кластера та між різними кластерами. Коефіцієнт Silhouette Score набуває значень від 1 до 1[11], де:

- значення, близьке до 1, означає добре відокремлені та компактні кластери;
- близьке до - 0 класи перетинаються;
- від'ємне - об'єкти можуть бути віднесені до неправильного кластера.

Модель перевірялась для к∈ [3; 6], результати виводились у термінал(рис. 2.14):

```
[k=3] silhouette=0.335
[k=4] silhouette=0.285
[k=5] silhouette=0.283
[k=6] silhouette=0.28
```

Рис. 2.14 Знаходження оптимального silhouette

Найвище значення спостерігалося при k = 3, що вказує на найкраще співвідношення між щільністю та роздільністю кластерів.

Метод ліктя (Elbow Method) базується на аналізі значення Inertia (SSE) суми квадратів відстаней між об'єктами та центроїдами кластерів.(рис. 2.15) Інерція завжди зменшується зі збільшенням кількості кластерів, проте після певного моменту темп зменшення різко сповільнюється. Ця "точка перегину" графіка (лікоть) вказує на оптимальне значення к, за якого подальше збільшення кількості кластерів не дає суттєвого покращення моделі[12]. У проведеному експерименті "лікоть" спостерігався в діапазоні k = 5-6.

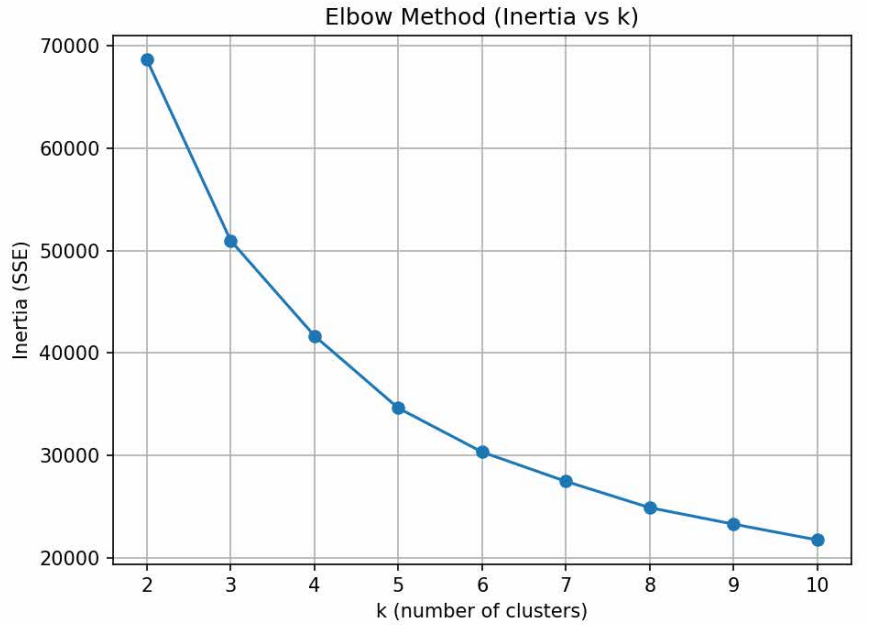

Рис 2.15 Метод ліктю

Порівняння обох методів показало, що хоча Elbow Method вказує на потенційне покращення при більшому k, показник Silhouette Score демонструє кращу якість кластеризації при меншій кількості груп. Остаточно було обрано кількість кластерів k = 3, оскільки саме при цьому значенні модель забезпечує оптимальний баланс між компактністю, роздільністю кластерів та інтерпретованістю результатів.

Навчання моделі K-Means. Після вибору параметра k виконується навчання фінальної моделі: Отримані мітки (labels) додаються до датафрейму у вигляді нового поля ClusterID_KMeans[13]. Після цього для кожного кластера обчислюються медіанні значення за основними ознаками (Distance_Km, Delivery_Time, Pace_MinPerKm, AgentRating) і підраховується розмір кластера (кількість записів).

Збереження результатів. Отриманы показники зберігаються у csv файл, в каталозі results для подальших операцій.

Підсумок. Метод кластеризації K-Means у даній системі використано для групування поставок за подібністю основних числових характеристик - відстані, тривалості, темпу та рейтингу кур'єра. Такий підхід дозволяє виявити природні закономірності у даних без попереднього визначення класів і служить інструментом для сегментації замовлень відповідно до їхніх поведінкових або логістичних параметрів.

Застосування K-Means забезпечує автоматичне формування кластерів, кожен з яких відображає певний тип доставки (швидкість, середні або повільні за часом виконання). Алгоритм виконує багатоетапну підготовку даних очищення, усунення викидів, масштабування та перевірку якості кластеризації через коефіцієнт силуету, що підвищує достовірність поділу та стабільність результатів. Отримана модель кластеризації відображає важливу роль у

подальшому аналізі, оскільки дозволяє розглядати кожен кластер як окрему групу спостережень із власними закономірностями. Надалі вона служить основою для порівняння ефективності маршрутів, формування профілів клієнтів та подальшої інтеграції з іншими алгоритмами Data Mining у межах систем підтримки прийняття рішень транспортної компанії.

4.8 Пошук асоціативних правил (Apriori)

Мета методу. Метою застосування методу Apriori є виявлення закономірностей між атрибутами доставок, які відбуваються одночасно або з високою ймовірністю. Отримані правила дозволяють виявити взаємозв'язки між умовами перевезення (район, транспорт, погода, рейтинг агента, тощо) та їхнім результатом (тривалістю доставки), що підвищує пояснювальність результатів класифікації та дозволяє глибше зрозуміти структуру логістичних проценсів. Таким чином, асоціативні правила допомагають керівництву транспортної компанії ідентифікувати приховані залежності між умовами перевезень і результатом доставки, що підвищує якість прийняття рішень[14].

Підготовка даних. Дані завантажуються з представлення dbo.vw_MiningInput у базі Amazon Delivery через ODBC Driver. З представлення відбираються наступні поля:

- Area – Район в якому виконувалась доставка;
- Vehicle – тип транспортного засобу;
- Category – категорія замовлення;
- Weather, Traffic – погодні та дорожні умови;
- DayPart – частина доби;
- RatingBand – діапазон рейтингу кур'єра;
- DeliveryTime – час виконання доставки.

Далі виконується категоризація часу доставки у змінну DelayBand за інтервалами: 0-60, 61-120, 120-180, 180+ хвилин. Ця ознака є цільовою складовою для аналізу залежностей між умовами доставки та її результатом.

Формування транзакцій (One-Hot Encoding). Кожен рядок у наборі даних розглядається як транзакція, а кожен атрибут – як "товар". Для цього усі текстові поля перетворюються у формат Колонка=Значення. Далі формується двійкова матриця (one-hot encoding), де кожна колонка відповідає певній категорії, а значення True або False вказує на наявність цієї властивості у транзакції. Таке подання дозволяє алгоритму Apriori працювати аналогічно аналізу "кошиків покупок", де кожна доставка є набором умов.

Пошук частих наборів і правил. На сформованих транзакціях застосовується алгоритм Apriori із пороговими значеннями :

- min_support = 0.05,
- min_confidence = 0.40,
- min_lift = 1.00.

Виявлені часті набори елементів передаються у функцію association_rules() з бібліотеки mlxtend.frequent_patterns для генерації асоціативних правил. Для кожного правила обчислюються метрики Support, Confidence, Lift, Leverage, Conviction і довжина правила (кількість елементів у Antecedent та Consequent). Отримані результати сортуються за значеннями Lift та Confidence, після чого зберігаються у файл results/rules.csv[15].

Інтерпретація результатів. Побудовані правила дозволяють встановити, які поєднання умов найбільш ймовірно ведуть до певного діапазону часу доставки (DelayBand). Завдяки цьому метод Apriori підсилює аналітичну складову системи Data Mining, забезпечуючи пояснювальний рівень для результатів класифікації та кластеризації. Такі закономірності можуть бути використані для формування рекомендацій щодо планування маршрутів і оцінки впливу зовнішніх факторів на ефективність перевезень[16].

4.9 Дослідження поведінки класифікаційної моделі Наївного Байєса в межах кластерів даних

Мета методу. Метою комбінування кластеризації з методом Наївного Байєса є дослідження стабільності та якості класифікаційної моделі в різних групах логістичних даних. Кластеризація дозволяє виділити сегменти доставок із подібними характеристиками (тип транспорту, погодні умови, частина доби тощо), а подальше застосування класифікації дає змогу оцінити, наскільки рівномірно модель працює в межах кожного кластера. Такий підхід не змінює логіку навчання, але забезпечує більш глибоке розуміння поведінки моделі та підвищує її пояснювальність.

Підготовка даних. Аналіз виконується на основі даних із уявлення vw_MiningInput у базі SQL Server, яке містить ключові поля таблиці фактів доставки (FactDeliveries) і пов'язаних вимірів (DimArea, DimVehicle, DimCategory, DimWeather, DimTraffic, DimTime, DimAgent). Для кожного запису формується розрахунковий атрибут Pace_MinPerkm середній час доставки однієї одиниці відстані, що дозволяє врахувати ефективність маршрутів різної довжини.

До моделі включаються числові ознаки:

- Distance_Km, Delivery_Time, Pace_MinPerKm, AgentRating;

та категоріальні:

- Area, Vehicle Category, Weather, Traffic, DayPart.

Цільова змінна - DelayBand, що формується на основі атрибуту Delivery_Time і поділяє замовлення на інтервали: 0-60, 61-120, 120-180, 180+ хвилин.

Реалізація комбінованого підходу. Побудована модель GaussianNB (реалізація методу Наївного Байєса) навчається на всьому обсязі даних із попередньою обробкою ознак у пайплайні, який включає:

- заповнення пропусків (SimpleImputer),

- стандартизацію числових полів (StandardScaler),
- кодування категоріальних ознак методом One-Hot Encoding.

Після глобального навчання модель оцінюється на тестовій вибірці, а отримані результати (y_pred, y_true) зберігаються разом із ключами доставок.
Далі ці дані об'єднуються з результатами кластеризації (kmeans_clusters.csv), що містить номери кластерів (ClusterID_KMeans) для кожної доставки.
Для кожного кластера обчислюються окремі метрики:

- accuracy,
- macro F1-score,
- кількість спостережень у кластері.

Отримані результати зберігаються у таблиці nb_global_cluster_scores.csv, де фіксуються середні, медіанні та зважені значення F1-метрики по всіх кластерах. Таким чином формується детальна аналітика щодо стабільності моделі в різних групах даних.

Підсумок. Розроблений модуль комбінування кластеризації та методу Наївного Байєса реалізує послідовну логіку дослідження логістичних даних. На першому етапі дані обробляються та кластеризуються методом K-Means для виокремлення груп маршрутів із подібними характеристиками. Далі побудована глобальна класифікаційна модель Наївного Байєса навчається на всій вибірці, після чого виконується оцінювання її роботи в межах кожного кластера. Така структура дає змогу реалізувати єдиний обчислювальний процес, що поєднує кластеризацію та класифікацію без зміни навчальної вибірки, забезпечуючи можливість подальшого дослідницького аналізу стабільності моделі у різних сегментах даних. Результати виконання зберігаються у вигляді таблиць з метриками точності та F1-оцінками, які в подальших розділах використовуються для аналітичного порівняння та інтерпретації поведінки моделі.

4.10 Пояснення результатів класифікації за допомогою асоціативних правил

Мета методу. Метою цього етапу є підвищення пояснювальності моделі Наївного Байєса шляхом порівняння її прогнозів із уже виявленими асоціативними правилами. Якщо класифікатор передбачає певну категорію затримки доставки (DelayBand), модуль перевіряє, чи існує відповідне правило Apriori, яке логічно пояснює цей прогноз через поєднання атрибутів (тип транспорту, район, погода, рейтинг виконавця тощо). Таким чином, система не лише прогнозує, але й аргументує свої рішення, що особливо важливо для практичного використання в управлінні логістичними процесами.

Підготовка даних. Програма підключається до бази Amazon Delivery через драйвер ODBC Driver 17 for SQL Server та завантажує дані з уявлення dbo.vw_MiningInput, яке містить сукупність показників доставки (час, відстань, рейтинг, погодні умови, транспорт, частину доби, категорію тощо).

У ході підготовки:

- обчислюється допоміжний показник Pace_MinPerkm середній темп доставки в хвилинах на кілометр;
- формується цільова змінна DelayBand за інтервалами 0-60, 61-120, 120-180, 180+ хвилин;
- здійснюється очищення від пропусків і кодування категоріальних ознак (One-Hot Encoding);
- для числових параметрів (Distance_Km, Delivery_Time, Pace_MinPerKm, AgentRating) застосовується імп'ютеризація за медіаною і стандартизація значень.

Формування та навчання моделі. Модель GaussianNB (Наївний Байєс із нормальним розподілом) навчається на 80% даних, а 20% використовуються для тестування. Після навчання обчислюються метрики accuracy і macro F1-score, які виводяться у консоль, а результати прогнозів для тестової вибірки зберігаються у файл nb_test_predictions.csv

Кожен запис тестової вибірки зберігає:

- ідентифікатор доставки (DeliveryKey або OrderID),
- категоріальні атрибути (Area, Vehicle, Category, Weather, Traffic, DayPart),
- рейтинг виконавця,
- та прогнозований клас PredDelayBand_NB

Порівняння з асоціативними правилами. Після формування прогнозів моделі Наївного Байєса виконується процедура порівняння їх із набором асоціативних правил, отриманих на попередньому етапі. Основна мета цього процесу - визначити, чи можна пояснити кожен прогноз класифікатора існуючим правилом, що відображає логічний зв'язок між умовами доставки та її результатом. На початку скрипт завантажує таблицю прогнозів моделі (з полями атрибутів доставки і передбаченим класом DelayBand) та таблицю асоціативних правил із полями Antecedent, Consequent, Confidence і Lift. Далі для кожного прогнозу модуль вибирає лише ті правила, у яких наслідок (Consequent) відповідає прогнозованому класу моделі. Серед цих правил перевіряється умова, щоб усі елементи передумови (Antecedent) містилися серед характеристик конкретної доставки. Якщо така відповідність виявлена, правило вважається релевантним до цього прогнозу. У разі наявності кількох відповідних правил вибирається найкраще за критеріями Confidence, Lift і кількістю умов у передумові. Для кожного прогнозу зберігається знайдене правило та його метрики у підсумковій таблиці nb_assoc_explanations.csv. Якщо жодне правило не підходить, запис позначається як непояснений. Таким чином, реалізований алгоритм виконує систематичне зіставлення між прогнозами класифікаційної моделі та виявленими закономірностями, формуючи пояснювальну частину Data Mining-модуля, яка показує, у яких випадках рішення моделі підкріплені логічними правилами, знайденими методом Apriori.

Підсумок. Розроблений модуль пояснення результатів класифікації реалізує механізм зіставлення прогнозів моделі Наївного Байєса з уже виявленими асоціативними правилами. Для цього використовується спільна структура даних, що містить прогнозовані класи та характеристики кожної доставки, а також набір правил із відповідними метриками довіри. Порівняння виконується за принципом відповідності наслідку правила прогнозованому класу та перевірки наявності всіх умов передумови серед атрибутів доставки. Отримані збіги формуються у таблицю пояснень, що зберігає найкраще правило для кожного прогнозу та його параметри (Confidence, Lift, MatchLen). Таким чином, модуль забезпечує технічну основу для побудови пояснювального шару системи Data Mining, який дозволяє інтерпретувати роботу класифікаційної моделі через знайдені закономірності між факторами доставки.

4.11 Виявлення асоціативних правил у межах кластерів даних

Мета методу. Метою даного етапу є виявлення локальних закономірностей між умовами виконання доставок у межах кожного окремого кластера, сформованого попереднім етапом кластеризації. Якщо глобальний пошук асоціативних правил показує загальні залежності у всій вибірці даних, то кластерний підхід дозволяє виявити специфічні взаємозв'язки, притаманні певним групам маршрутів або категоріям замовлень. Такий підхід підвищує точність інтерпретації, оскільки кожен кластер має власні особливості наприклад, доставку певного типу товарів, виконану в схожих погодних умовах або в конкретних районах міста. Виявлені закономірності допомагають зрозуміти, які чинники в кожній групі найбільше впливають на час виконання доставки.

Загальна логіка роботи модуля. Розроблений модуль отримує на вхід два основних набори даних: результати кластеризації у вигляді таблиці з ідентифікаторами доставок і номерами кластерів, а також аналітичне уявлення vw_MiningInput з бази SQL Server, яке містить ключові атрибути кожної доставки район виконання, тип транспортного засобу, категорію замовлення, погодні та дорожні умови, частину доби, рейтинг кур'єра та час виконання доставки. Після завантаження обидва джерела об'єднуються за унікальним ключем доставки. У результаті кожна доставка отримує не лише набір власних характеристик, а й кластерну належність, що дозволяє аналізувати закономірності окремо для кожної групи. У ході роботи скрипт послідовно проходить усі кластери, формує для кожного з них окрему вибірку та виконує побудову наборів транзакцій, які потім обробляються алгоритмом Apriori. Таким чином, створюється незалежний процес аналізу для кожного кластера, що гарантує ізольованість результатів і відсутність впливу інших груп на локальні закономірності. Такий підхід є більш гнучким і точним, адже дозволяє

порівнювати структуру знань між кластерами, визначати, у яких групах зв'язки між атрибутами є найстійкішими, а де взаємозалежності менш очевидні.

Підготовка транзакцій. На етапі підготовки даних кожен запис перетворюється у вигляд набору категоріальних ознак, що описують умови конкретної доставки. Для цього усі атрибути переводяться у формат «ключ-значення»: Area=Center, Vehicle=Motorcycle, Weather=Sunny, Traffic=High. Така форма представлення дозволяє уніфікувати всі дані та забезпечити сумісність із алгоритмом Apriori, який аналізує частоту появи спільних елементів у множині транзакцій. Крім базових характеристик, у процес включаються дві додаткові категоріальні змінні. Перша RatingBand, яка розподіляє рейтинги кур'єрів на три рівні якості (високий, середній, низький), що дозволяє врахувати вплив професійності виконавця на результати. Друга — DelayBand, яка відображає інтервал фактичного часу виконання доставки: 0-60, 61-120, 120-180 або понад 180 хвилин. Цей атрибут виступає цільовим при побудові правил, оскільки саме він визначає результат виконання логістичної операції. У результаті формується набір транзакцій, де кожна доставка представлена комбінацією своїх властивостей. Ці транзакції групуються за кластерами, після чого для кожного кластера створюється власна матриця спостережень, придатна для аналізу методом Apriori.

Пошук правил у межах кластерів. Після підготовки даних модуль переходить до пошуку закономірностей усередині кожного кластера. Для цього застосовується алгоритм Apriori з динамічним визначенням мінімального значення підтримки (support). Це означає, що для малих кластерів поріг підтримки автоматично зменшується, аби не втратити потенційно цінні, але рідкісні комбінації ознак, тоді як для великих кластерів він збільшується для уникнення надлишку неінформативних правил. Такий підхід дозволяє досягти балансу між обсягом знайдених правил та їхньою достовірністю. Після виконання Apriori усі знайдені правила фільтруються таким чином, щоб у частині наслідку залишались лише ті, що містять змінну DelayBand. Таким чином, система відсікає всі другорядні взаємозв'язки й концентрується лише на закономірностях, що пояснюють поведінку часу доставки. Наприклад, у кластері, який об'єднує короткі міські маршрути, можуть бути знайдені правила виду Vehicle=Bike, Traffic High → DelayBand=120–180, що вказують на негативний вплив заторів саме для цього типу перевезень.

Формування результатів. Після обробки кожного кластера результати зберігаються у вигляді таблиць, шо містять повну інформацію про знайдені правила – передумови, наслідки, рівень підтримки, довіру (confidence) та

показник (lift). Окремо формується узагальнена таблиця з кількістю знайдених правил у кожному кластері.

Підсумок. Розроблений модуль дозволяє автоматично визначати специфічні закономірності у межах кожного кластера даних, що дає змогу не лише деталізувати загальні результати аналізу, але й виявити унікальні фактори, притаманні окремим групам доставок. Поєднання кластеризації з пошуком асоціативних правил підвищує глибину аналітичного процесу та формує новий рівень пояснюваності системи. Завдяки цьому створюється основа для подальшого порівняльного аналізу кластерів, визначення найбільш ризикових умов доставки та побудови більш обґрунтованих управлінських рішень у транспортній логістиці.

5 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

5.1 Апаратні та програмні вимоги

Розробка, тестування та аналітичні експерименти системи підтримки прийняття рішень проводились у середовищі Microsoft SQL Server та Visual Studio 2022, що забезпечують повний цикл реалізації сховища даних, побудови OLAP-кубів, KPI-показників, Data Mining-моделей і звітів. Для обробки даних також використовувались інструменти Microsoft Excel та Power BI для візуалізації та додаткової перевірки результатів. Архітектура розробленої системи базується на технологічному стеку Microsoft BI Platform, який охоплює кілька взаємопов'язаних компонентів:

1. Microsoft SQL Server Management Studio (SSMS) створення, адміністрування та наповнення бази даних «DataWarehouse_Logistics».
2. SQL Server Integration Services (SSIS) — етапи завантаження даних (ETL-процес), очищення, нормалізації та перенесення до сховища.
3. SQL Server Analysis Services (SSAS) — побудова багатовимірного OLAP-куба, створення мір, ієрархій та KPI.
4. SQL Server Reporting Services (SSRS) - розробка аналітичних звітів і дашбордів для управлінського аналізу.
5. Power BI Desktop — виконання експериментів Data Mining, кластеризації та візуального представлення результатів.
6. Python 3.13 (Visual Studio Code) — реалізація алгоритмів K-Means, Naïve Bayes і Apriori, а також розрахунок метрик точності та візуалізація кластерів.
7. Microsoft Excel 2019 — додаткові перевірки обчислень і формування допоміжних таблиць для експериментів.

Розроблена система функціонує як єдиний аналітичний комплекс. Дані проходять шлях: OLTP джерела – ETL (SISS) – DWH (SQL Server) – OLAP куб (SSAS) – Data Mining (Python/Power BI) – звітність (SSRS/Power BI). Такий підхід забезпечує узгодженість даних, масштабованість аналітики й можливість одночасного використання описової, діагностичної та прогнозової аналітики в межах однієї системи підтримки прийняття рішень.

5.2 Результати виконання методу Наївного Байєсу

Застосування методу Наївного Байєсу у межах системи підтримки прийняття рішень було спрямоване на побудову багатокласової моделі, яка класифікує затримки доставки за часовими інтервалами. Метою експерименту було виявлення статистичних закономірностей між умовами доставки та рівнем її своєчасності, а також перевірка того, наскільки цей підхід здатен коректно розпізнавати приховані шаблони у великих обсягах даних логістичної системи.

Для навчання моделі використовувалося представлення vw_MiningInput, сформоване на основі фактичної таблиці FactDeliveries та вимірів, що характеризують середовище виконання доставки. Цільову змінну створено шляхом поділу загального часу доставки на чотири класи: 0-60 хвилин, 61-120 хвилин, 120-180 хвилин та понад 180 хвилин. Така розбивка дозволила не лише відстежувати факт запізнення, а й визначити ступінь його впливу. Для підвищення пояснювальної сили моделі додатково було розраховано похідний показник Pace_MinPerKm, який описує середню швидкість доставки на кілометр маршруту. Розрахунки виконувались у середовищі Python із використанням бібліотеки scikit-learn. Модель навчалася на 80% вибірки, а решта 20% використовувалася для тестування. Попередня обробка ознак включала імп'ютацію пропущених значень, стандартизацію числових полів та кодування категоріальних змінних у вигляді One-Hot Encoding. Навчання проводилося за алгоритмом Gaussian Naive Bayes, який є оптимальним для даних із числовими ймовірнісними розподілами та великою кількістю незалежних ознак. Результати тестування моделі продемонстрували точність 0.814 та середній макро-F1 показник 0.776(рис. 3.1), що вказує на достатньо високу якість класифікації для багатокласової задачі. Додатковий аналіз результатів проведено на основі трьох візуалізацій: матриці помилок, порівняння розподілу прогнозованих і фактичних класів, а також теплової карти середніх значень ознак для кожного класу.

Рис. 3.1 Точність та F1 моделі

Матриця помилок показала(рис. 3.2), що модель найкраще розпізнає класи 61-120 хв та 120-180 хв, де кількість правильних передбачень перевищує три тисячі випадків. Для цих двох категорій модель демонструє чітке розмежування і стабільність, що пояснюється наявністю великої кількості спостережень і відносно однорідними характеристиками доставок. У класі 0-60 хв (своєчасна доставка) спостерігається певна кількість помилкових передбачень як 61-120 хв, що свідчить про близькість параметрів між двома межовими групами. Найбільше перехресних помилок фіксується між класами 120-180 та 180+, де значна частина випадків із довгими затримками була віднесена до менш критичних категорій. Це зумовлено високою варіативністю впливових факторів, зокрема погодних умов, типу транспорту та густоти трафіку.

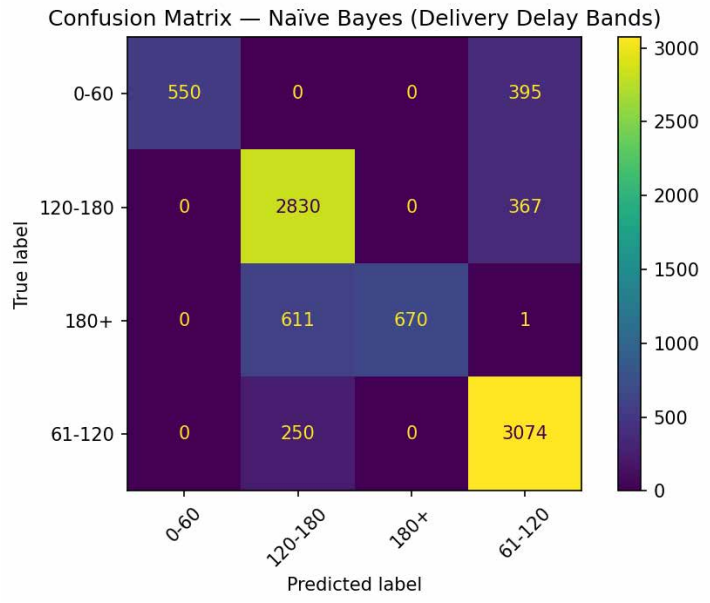

Рис 3.2 матриця помилок NB

Додаткова візуалізація розподілу фактичних та прогнозованих класів(рис. 3.3) підтвердила стабільність роботи моделі на рівні загальної структури даних. Пропорції кожної групи у прогнозі практично збігаються з реальними частками вхідного набору, що свідчить про відсутність систематичних зсувів у бік конкретного класу. Таким чином, модель не лише демонструє високу середню точність, але й зберігає адекватну збалансованість між частими та рідкісними класами, що є важливим для управлінських рішень, заснованих на прогнозуванні рівня запізнення.

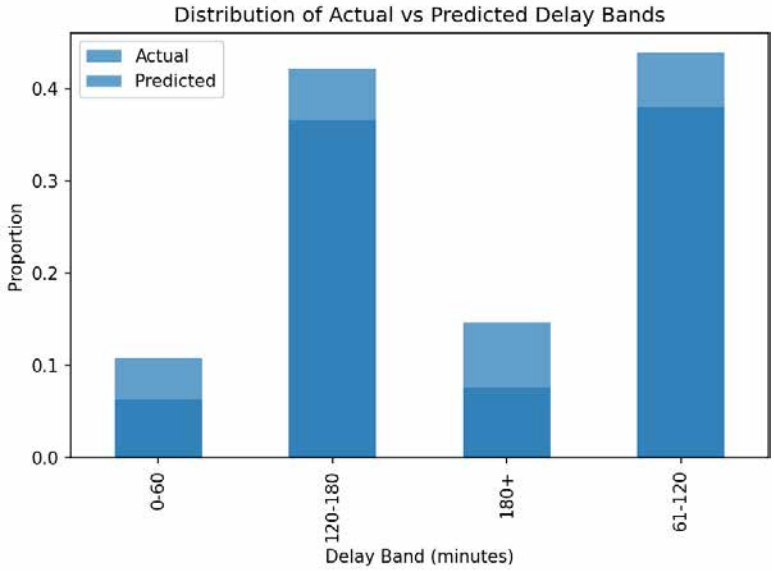

Рис. 3.3 Стовпчикова діаграма знайдених класів

Найбільш глибоке уявлення про логіку роботи алгоритму надає теплова карта середніх значень ознак GaussianNB Per-Class Feature Means(рис. 3.4), Вона показує, як середні стандартизовані значення кожної ознаки відрізняються між класами затримок. З аналізу карти видно, що для класу 120-180 хв домінують ознаки, пов'язані з підвищеною відстанню та більш складними транспортними

умовами. У класі 180+ хв спостерігається чітке зростання середніх значень Distance_Km та Delivery_Time, а також виражений вплив категоріальних змінних - зокрема, підвищена частка доставок у районах типу Semi-Urban, за несприятливої погоди (Stormy або Windy) та високого трафіку (Traffic_High або Traffic_Jam). Також простежується кореляція між великим транспортом (Vehicle_Van, Vehicle_Truck) і ймовірністю потрапляння у класи з тривалими затримками. Натомість невеликі засоби пересування, такі як scooter або bicycle, переважають у класах швидких доставок.

Інтерпретація цих спостережень підтвердила гіпотезу про наявність чітких факторних закономірностей: погодні умови, тип транспортного засобу, відстань та інтенсивність трафіку є ключовими предикторами часу доставки. Алгоритм Наївного Байєса дозволяє оцінити не лише ймовірність затримки, а й виявити, які саме поєднання ознак зумовлюють її виникнення. Це робить його корисним інструментом не тільки для прогнозування, але й для пояснювальної аналітики, яка є критично важливою для управлінських рішень у логістичних компаніях.
Таким чином, підсумовуючи результати експерименту, можна відзначити, що модель Наївного Байєса забезпечує стабільну класифікацію рівнів затримок із точністю понад 81%. Вона адекватно відображає структуру даних, демонструє низький рівень систематичних похибок і високу інтерпретованість. Отримані результати використовуються як еталонна модель точності для подальших порівнянь з іншими підходами, зокрема кластеризацією та комбінованими аналітичними схемами, що розглядаються у наступних підрозділах.

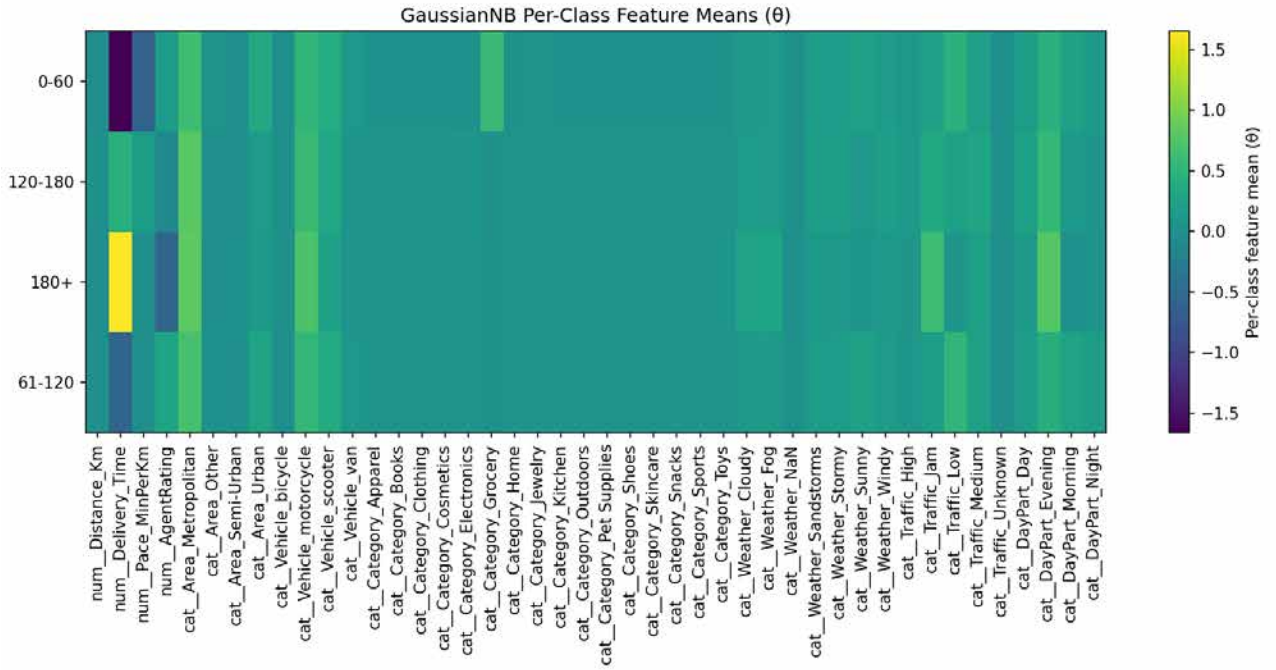

Рис. 3.4 Теплова карта ознак

5.3 Результати кластеризації

Метою кластеризації у межах побудованої системи було виявлення прихованих груп доставок, що характеризуються схожими умовами виконання та поведінковими особливостями. Такий підхід дозволяє вийти за межі прямої класифікації і сформувати більш глибоке розуміння структури даних, виявивши природні поділи у вибірці без попереднього знання про мітки класів. Це, у свою чергу, дає можливість оптимізувати планування маршрутів, навантаження транспортних засобів і прогнозування ризиків затримок.

Кластеризація виконувалася методом K-Means, який забезпечує компактність груп за ознакою мінімізації внутрішньокластерної варіації. Вибір цього методу обґрунтовано його ефективністю для числових багатовимірних даних та зрозумілістю результатів, що є важливим для управлінських рішень. Перед початком аналізу всі числові змінні було нормалізовано, а категоріальні закодовано за допомогою One-Hot Encoding. У модель включено ключові фактори, що мають найбільший вплив на доставку: Delivery_Time, Distance_Km, Pace_MinPerKm, AgentRating, Traffic_Level, Weather, Vehicle_Type та Category. Для визначення оптимальної кількості кластерів застосовано комбінацію методу «лікоть» (Elbow Method) та аналізу силуетного коефіцієнта (Silhouette Score). За результатами було обрано значення k = 3, що дозволило виділити три типові кластери доставок. Обраний силует-скор становив 0.335, що свідчить про виражену структуру груп та низький рівень перекриття між ними. На графіку(рис. 3.5) відображено взаємозв'язок між часом доставки, відстанню маршруту та рейтингом агента для трьох знайдених кластерів. Усі центроїди (позначені жовтим маркером) утворюють просторову градацію, яка узгоджується зі зміною рівня ефективності виконання доставки.

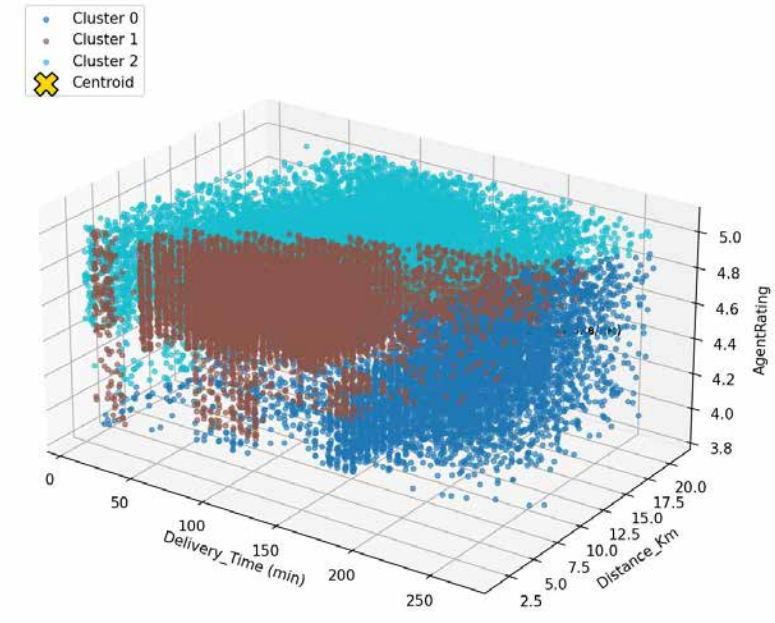

Рис. 3.5 Візуалізація кластерів

Перший кластер (Cluster 0) включає 6774 спостереження і характеризується високими значеннями часу доставки (≈185 хв) та значною дистанцією (≈12,4 км) при відносно невисокому рейтингу агентів (≈4,3). Це типові довгі або складні маршрути, які часто пов'язані з підвищеним трафіком чи несприятливими погодними умовами. Для цих доставок також спостерігається вищий темп у хвилинах на кілометр (≈15,1), що свідчить про меншу швидкість пересування. Таким чином, кластер 0 відображає групу низькоефективних або ризикових рейсів, які вимагають оптимізації.

Другий кластер (Cluster 1) об'єднує 10950 записів і має середній час доставки близько 120 хвилин при мінімальній дистанції (≈4,7 км). Його відмінною рисою є найвищий середній рейтинг агентів (≈4,8), проте середній темп виконання (≈21,8 хв/км) вказує на знижену швидкість у коротких міських поїздках, де затримки можуть бути спричинені високою щільністю руху або частими зупинками. Цей кластер можна визначити як операційно середній сегмент, у якому доставлення виконуються стабільно, але з потенціалом для підвищення швидкості.

Третій кластер (Cluster 2) є найчисельнішим — 18815 спостережень, із середнім часом виконання 105 хвилин та дистанцією 12,2 км. При цьому спостерігається найнижчий темп (≈8,7 хв/км), що означає високу швидкість пересування. Рейтинг агентів у цій групі також залишається високим (≈4,8). Цей кластер репрезентує найефективніші доставки, де завдяки оптимальним маршрутам, сприятливим умовам і досвідченим виконавцям забезпечується короткий час транспортування навіть на довших дистанціях.

Загальний аналіз медіанних(рис. 3.6) значень дозволив побачити чітку залежність між тривалістю виконання, відстанню, середнім темпом та оцінкою агентів. Для кластерів 0 і 1 збільшення часу доставки пов'язане зі зменшенням рейтингу та підвищенням темпу (тобто зниженням ефективності), тоді як у кластері 2 простежується зворотна тенденція. Це свідчить про те, що рейтинг агентів виступає індикатором дисципліни виконання маршруту і водночас корелює з параметрами продуктивності. Отримані результати підтверджують, що алгоритм K-Means адекватно відтворює внутрішню структуру логістичної системи, формуючи три чітко інтерпретовані сегменти: ефективні (Cluster 2), збалансовані (Cluster 1) та ризикові (Cluster 0) доставки.

```
 ClusterID_KMeans
0     6774
1    10950
2    18815
Name: count, dtype: int64

Медіани фіч по кластерах:
                Distance_Km  Delivery_Time  Pace_MinPerKm  AgentRating
ClusterID_KMeans
0                    12.393          185.0         15.124          4.3
1                     4.672          120.0         21.755          4.8
2                    12.218          105.0          8.717          4.8
```

Рис. 3.6 кількість значень в кластерах та медіанні значення.

Така сегментація створює основу для комбінованого аналізу, коли подальші методи (зокрема Наївний Байєс або пошук асоціативних правил) використовуються для оцінювання поведінки моделі у різних умовах, що дає змогу перевірити, чи зберігає Наївний Байєс свою стабільність і здатність до узагальнення незалежно від природи доставок. Такий підхід підвищує пояснюваність системи, оскільки результати моделі можна інтерпретувати окремо для кожного типу сценаріїв. Практичне значення кластеризації полягає у можливості гнучко застосовувати різні управлінські стратегії до кожної групи. Для ефективних кластерів доцільно підтримувати поточну організацію процесів, тоді як для ризикових рейсів система може ініціювати додаткові попередження або оптимізаційні рекомендації наприклад, зміну маршруту, часу відправлення чи виконавця. Таким чином, кластеризація відіграє роль аналітичного шару прогнозної підтримки, що дозволяє не лише спостерігати за минулими затримками, а й активно попереджати їх виникнення у майбутньому.

5.4 Результати асоціативних правил

Для підвищення пояснюваності роботи системи та виявлення прихованих зв'язків між параметрами доставки було застосовано метод пошуку асоціативних правил, реалізований за допомогою алгоритму Apriori. На відміну від класифікації або кластеризації, цей підхід не прогнозує результат, а виявляє закономірності співіснування атрибутів у даних. Таким чином, його основна мета полягає у формуванні інтерпретованих знань про поведінку логістичної системи, що можуть бути використані для управлінських рішень. Для аналізу використовувався узагальнений набір даних, який містив категоріальні ознаки: Category, RatingBand, DelayBand, Traffic, Weather, Area, DayPart та Vehicle. Кожне правило має структуру виду Antecedent → Consequent, де ліва частина (Antecedent) описує умови або ситуацію, а права (Consequent) пов'язаний із нею результат. Для оцінки значущості правил застосовувалися три метрики: Support

(частота спільного виникнення), Confidence (умовна ймовірність появи правої частини при виконанні лівої) та Lift (індикатор сили зв'язку між ознаками, що перевищує випадкову кореляцію).

Після фільтрації малозначущих комбінацій було згенеровано 828 асоціативних правил, які збережено у файлі resultsWrules.csv. Подальший відбір проводився за критерієм максимальної довіри (Confidence > 0.6) та високого значення Lift, що дозволило виокремити лише статистично значущі зв'язки. На bar chart(рис. 3.7) представлено десять найсильніших правил за показником Lift. Найбільш вираженою закономірністю виявилось поєднання Категорія = grocery → DelayBand = 0–60, що свідчить про стабільне виконання доставок продуктів харчування в межах однієї години.

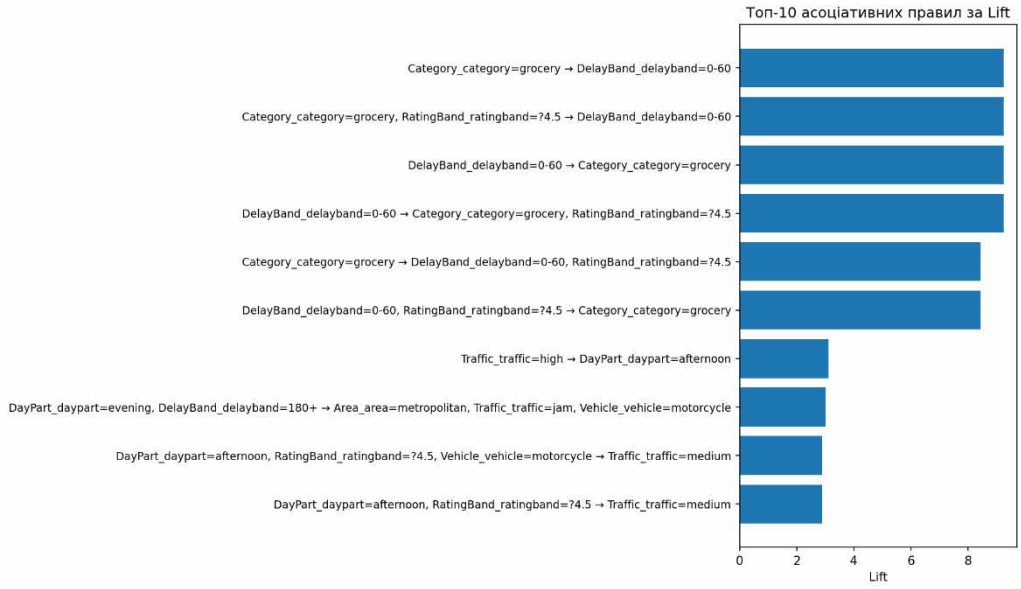

Рис. 3.7 Топ правил за Lift

Підвищення Lift для цього правила свідчить про те, що короткі доставки значно частіше трапляються саме для категорії "grocery", ніж можна було б очікувати випадково. Подібна закономірність зберігається й для правила Category = grocery, Rating Band ≥ 4.5 → DelayBand = 0–60, що вказує на взаємозалежність високого рейтингу агентів та швидкого виконання замовлень у продовольчому сегменті. Ці результати підкреслюють, що рейтинг агентів і тип товару є взаємопідсилюючими факторами для зменшення часу доставки.

Інші правила також мають логічну інтерпретацію:

- Traffic = high → DayPart = afternoon описує типову кореляцію між денним часом і піковим навантаженням доріг;

- DayPart = evening, DelayBand = 180+ → Area = metropolitan, Vehicle = motorcycle показує, що навіть при значних затримках у вечірній час в мегаполісах частіше використовуються мотоцикли як транспорт із високою мобільністю.

На діаграмі scatter plot(рис. 3.8) показано співвідношення показників Support і Confidence для всіх правил, де колір маркера відповідає значенню Lift. Найщільніша область розташована в діапазоні Support = 0.05-0.15 та Confidence = 0.6-0.9, що свідчить про наявність значної кількості частих, але помірно сильних правил. Найцінніші закономірності розташовані у верхній частині діаграми вони характеризуються високим рівнем довіри й помітним відривом Lift, що підтверджує реальну кореляцію між факторами, а не випадкову спільність.

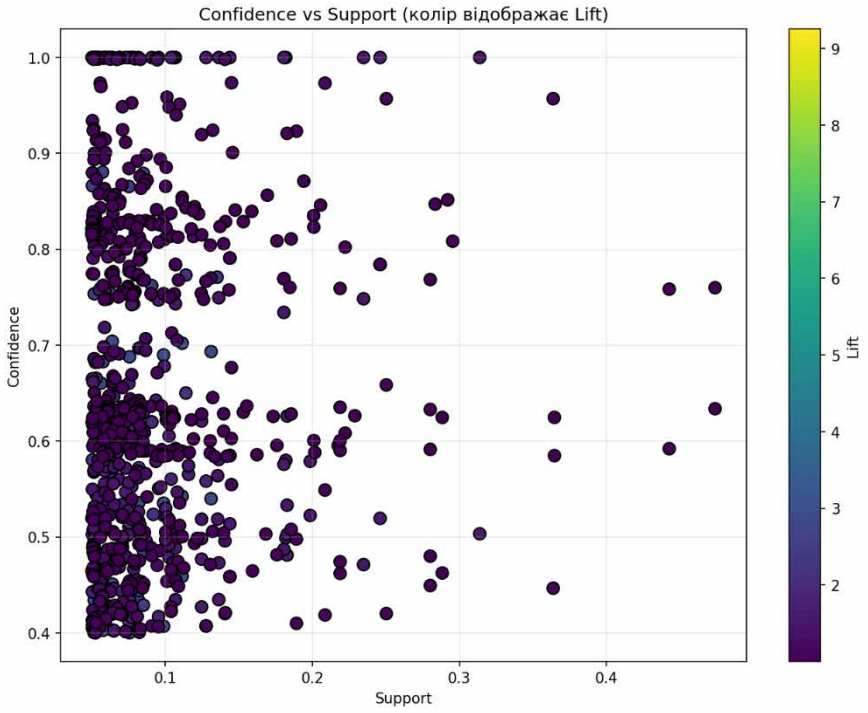

Рис. 3.8 Точкова діаграма співвідношення Support та Confidence

Для глибшого розуміння структури взаємозв'язків між ключовими атрибутами було побудовано теплову карту частоти співіснування токенів Antecedent → Consequent(рис. 3.9). Ця візуалізація є узагальненим підсумком роботи алгоритму Apriori, що відображає, наскільки часто певні фактори (Antecedent) з'являються в асоціативних правилах разом із пов'язаними наслідками (Consequent). На відміну від попередніх діаграм, які показували окремі приклади правил, дана теплова карта узагальнює їх у вигляді матриці частотності, дозволяючи оцінити щільність і напрямок взаємодії між різними змінними системи. По вертикалі відображені Antecedent–токени (фактори, що виступають

передумовами правил), а по горизонталі — Consequent-токени (атрибути, які найчастіше є наслідками). Колірна шкала праворуч позначає кількість входжень правил, у яких зустрічається відповідна пара Antecedent–Consequent. Чим інтенсивніший відтінок, тим частіше певна комбінація фігурує у знайдених закономірностях.

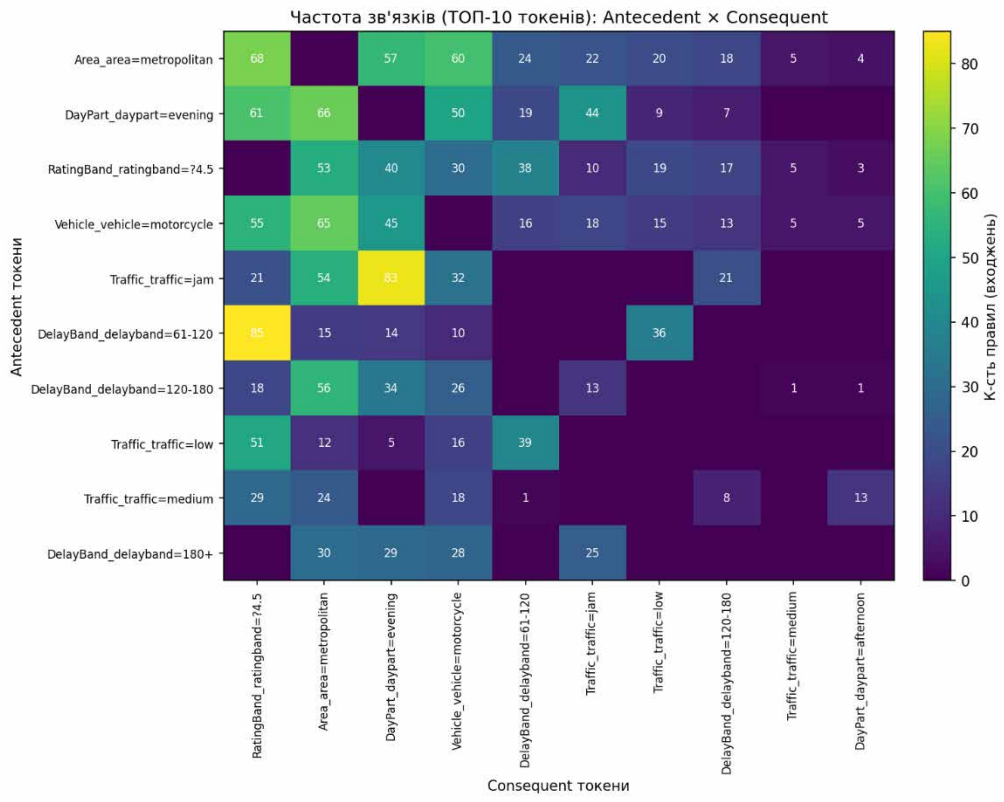

Рис. 3.9 Теплова карта передумов та наслідків

З аналізу карти видно кілька ключових закономірностей:

- DelayBand = 61–120 → RatingBand ≥ 4.5 (85). В асоціативних правилах проміжок затримки 61–120 хвилин часто супроводжується появою токена високого рейтингу агента. Це підкреслює, що в сегменті «середніх» затримок визначальними є зовнішні умови виконання, тоді як розподіл рейтингів не зміщується у бік низьких значень. Інакше кажучи, підвищена тривалість у цьому діапазоні не є індикатором слабкої якості роботи виконавця.
- Traffic = jam → DayPart = evening (83). Наявність «jam» найчастіше пов'язана саме з вечірнім часовим інтервалом. Розподіл правил фіксує стале співіснування цих токенів: під час вечірнього піку дорожня мережа перебуває у перевантаженому стані, що й відображається у частоті зв'язку.

- Area = metropolitan → RatingBand ≥ 4.5 (68). Міський ареал супроводжується появою високого рейтингу агента значно частіше, ніж випадково очікувалося б. Це означає, що у великі міста системно спрямовується більша частка виконавців з високими оцінками, і цей патерн стабільно фіксується правилами.
- DayPart = evening → Area = metropolitan (66). Вечірній часовий інтервал найчастіше поєднується з міськими локаціями. У структурі правил це проявляється як характерна пара «evening → metropolitan», що віддзеркалює концентрацію попиту у мегаполісах у кінці дня.
- Vehicle = motorcycle → Area = metropolitan (65). Мотоцикли в правилах найчастіше «ведуть» до міського ареалу. Така концентрація свідчить, що саме в метрополітенах цей тип транспорту є типовим інструментом виконання доставок.
- DayPart = evening → Rating Band ≥ 4.5 (61). У вечірніх слотах частіше фігурує токен високого рейтингу агента. Модель правил фіксує систематичне поєднання «evening» з «>4.5», що узгоджується з ускладненими умовами вечірнього періоду та потребою в більш досвідчених виконавцях.
- Area = metropolitan → Vehicle = motorcycle (60). Для міського середовища характерна підвищена частота появи мотоциклів як наслідку. Це симетрично підсилює попередню тезу: і «motorcycle → metropolitan», і «metropolitan → motorcycle» входять до найчастіших пар.
- Area = metropolitan → DayPart = evening (57). Міський ареал у правилах часто поєднується саме з вечірнім часом доби. Це означає, що для metropolitan вечірній інтервал є типовим контекстом появи залежностей з іншими токенами.
- DelayBand = 120–180 → Area = metropolitan (56). Затримки 120–180 хвилин у значній частці правил асоціюються з міськими зонами. Порівняно з іншим контекстом, саме «metropolitan» частіше виступає наслідком для цього діапазону затримок.
- Vehicle = motorcycle → RatingBand ≥ 4.5 (55). Використання мотоциклів нерідко супроводжується появою токена високого рейтингу. Це фіксує систематичний зв'язок між типом ТЗ та профілем виконавця в даних правил.

- Traffic = jam → Area = metropolitan (54). Токен «jam» у більшості випадків співіснує з міськими локаціями. Частота підтверджує, що перевантажений рух у вибірці — переважно міське явище.
- RatingBand ≥ 4.5 → Area = metropolitan (53). Високий рейтинг часто «вказує» на metropolitan як середовище виконання. У сукупності з попередніми парами це формує стабільний двонапрямний контур «rating ≥4.5 metropolitan».
- Traffic = low → RatingBand ≥ 4.5 (51). При низькому трафіку частіше фіксується високий рейтинг виконавця як наслідок у правилах. Це свідчить, що «легкий» дорожній контекст у вибірці не пов'язаний зі зміщенням до низьких оцінок; навпаки, спостерігається регулярне співіснування з високими.
- DayPart = evening → Vehicle = motorcycle (50). Вечірній період стабільно поєднується з вибором мотоцикла у ролі транспортного засобу. Це завершує трикутник частих співвідношень «evening ↔ metropolitan motorcycle», який теплокарта демонструє як домінантний сценарій у даних.

Теплова карта також виявила асиметрію між Antecedent і Consequent токенами. Ознаки, що описують зовнішні умови (як-от Traffic або Area), частіше виступають у ролі Antecedent, тоді як результативні показники (RatingBand, DelayBand, DayPart) переважно з'являються у ролі Consequent. Це підтверджує логічну спрямованість правил від факторів впливу до наслідків поведінки системи. Важливо, що така форма представлення даних дає змогу не лише бачити окремі закономірності, а й ідентифікувати найбільш впливові вузли взаємодії у логістичній системі. Зокрема, комбінації "Area-DelayBand" та "Traffic-DayPart" можна вважати домінантними кластерами поведінкових залежностей, які визначають типові сценарії доставки. Таким чином, теплова карта частоти зв'язків стала ключовим інструментом інтерпретації асоціативних правил, оскільки дозволила перейти від індивідуального аналізу правил до їх узагальненого статистичного профілю. Вона наочно демонструє, як саме різні фактори взаємодіють між собою у системі доставки, формуючи типовий контекст ефективної або, навпаки, ризикової логістичної поведінки.

Підсумовуючи, результати аналізу асоціативних правил показали, що в системі простежується низка стабільних і логічно узгоджених залежностей, які узгоджуються з попередніми висновками класифікаційного та кластерного аналізу. Отримані знання стали важливим елементом комплексної пояснюваної аналітики, що забезпечує глибше розуміння функціонування логістичних процесів і підтримує прийняття рішень на основі даних.

5.5 Результати дослідження поведінки класифікаційної моделі Наївного Байєса в межах кластерів даних

У цьому підрозділі наведено результати оцінювання роботи класифікаційної моделі Наївного Байєса в межах кластерів, отриманих попередньо методом K-Means. Такий підхід дозволяє дослідити стабільність і варіативність точності класифікації в різних групах логістичних даних, що відрізняються характеристиками маршрутів, тривалістю виконання доставок, рейтингом агентів та умовами перевезення. Після виконання кластеризації всі спостереження були розділені на три кластери, для кожного з яких окремо розраховано показники точності (Accuracy) та макросередньої F1-міри (Macro F1-score)(рис. 3.10).

	ClusterID_KMeans	size	accuracy	macro_f1
0	0	4667	0.825584	0.703368
1	1	1855	0.743396	0.784445
2	2	1757	0.862834	0.608499

Рис. 3.10 Поведінка моделі у різних кластерах

Отримані значення свідчать, що класифікаційна модель зберігає високий рівень узагальнювальної здатності в усіх групах даних. Незважаючи на варіацію між кластерами, розкид показників є помірним, що підтверджує стабільність алгоритму Наївного Байєса при зміні структури вхідних даних.

Аналіз результатів у межах кластерів.

Кластер 0 - "Середньо-швидкі доставки". Цей кластер є найбільшим за обсягом (понад 4600 записів) і характеризується середніми значеннями часу виконання та темпу руху. Модель демонструє точність 82,6%, при цьому F1 = 0,70, що свідчить про збалансовану роботу між основними класами. На графіку розподілу класів(рис. 3.11) спостерігається помірна перевага середніх затримок (61-120 хв), які модель ідентифікує найточніше. Короткі (0-60 хв) та наддовгі (180+) доставки частково змішуються через схожі комбінації умов, однак їхня частка незначна, тому це не впливає на загальний результат.

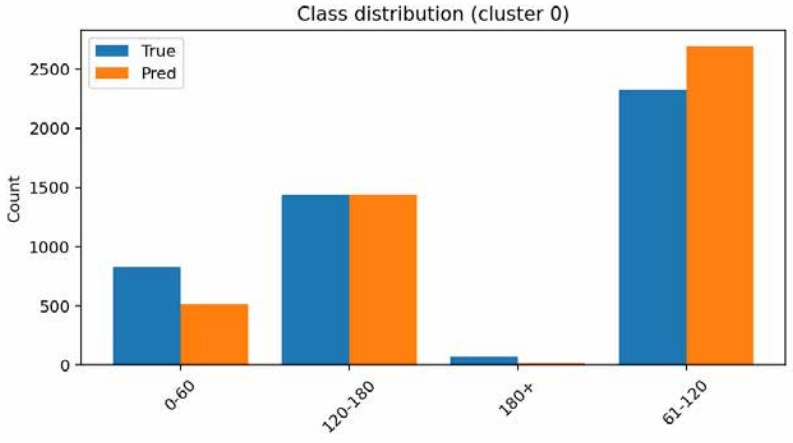

Рис. 3.11 Розподіл передбачених класів у кластері 0

Кластер 1 – "Довгі маршрути / нестабільні умови". Кількість спостережень у цьому кластері менша (≈1850), однак саме тут модель демонструє(рис. 3.12) найвищий баланс між точністю та повнотою: F1 = 0,78 при Accuracy = 0,74. Незважаючи на нижчу абсолютну точність, саме цей кластер показує найкращу здатність класифікатора адекватно розподіляти спостереження між усіма чотирма класами. Ймовірно, це зумовлено більш рівномірним розподілом факторів, що впливають на час доставки — тип транспорту, трафік і погодні умови. Кластер 1 можна охарактеризувати як "збалансований", де модель не переважає жодного класу, що важливо для загальної узгодженості прогнозів.

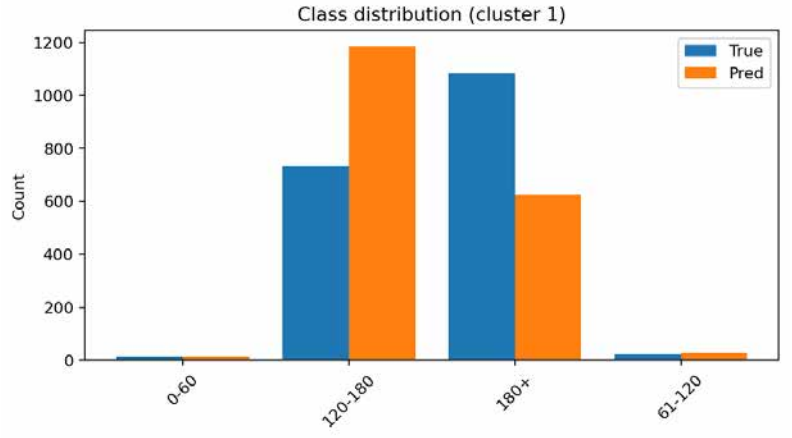

Рис. 3.12 Розподіл передбачених класів у кластері 1

Кластер 2 – "Швидкі та короткі маршрути". Найменший за розміром кластер (≈1750 записів) демонструє найвищу точність 86,3%, але при цьому має найнижчу F1-міру (0,61). Такий результат(рис. 3.13) свідчить про нерівномірність розподілу класів: модель надмірно добре розпізнає основний домінантний клас ("61–120 хв"), але гірше працює з крайніми інтервалами ("0–60" і "180+"). На графіках метрик по класах видно, що Recall для найшвидших доставок нижчий за середній рівень, що зумовлено меншою кількістю прикладів у тренувальній вибірці.

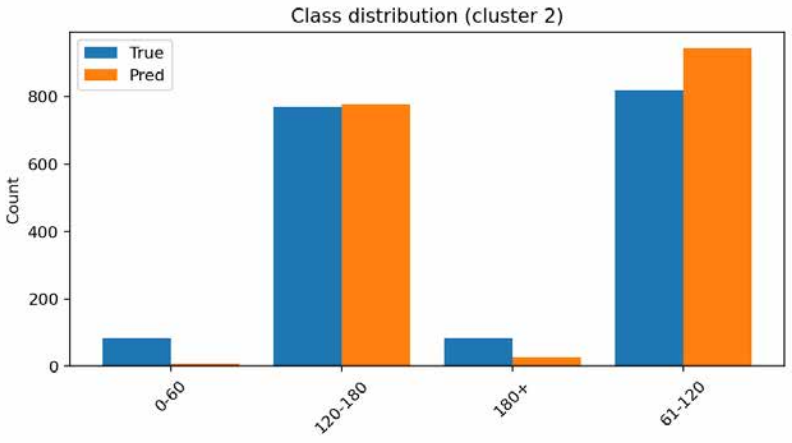

Рис. 3.13 Розподіл передбачених класів у кластері 2

Для оцінювання якості роботи моделі в межах кожного кластера побудовано графіки покласових метрик precision, recall та F1-score. Вони відображають ступінь відповідності прогнозів реальним даним для кожної з чотирьох категорій часу доставки: 0-60 хв, 61-120 хв, 120-180 хв та 180+ хв.

Кластер 0. Для першого кластера спостерігається(рис. 3.14) чітка перевага класів середньої тривалості доставок (61–120 та 120–180 хв), де значення усіх трьох метрик перевищують 0.8. Це свідчить про добре навчений класифікатор у межах домінантних класів. Для коротких доставок (0–60 хв) значення recall нижче (≈0.6), що вказує на втрату частини позитивних прикладів. Найгірше модель працює із тривалими доставками (180+ хв), де recall опускається нижче 0.3, що свідчить про сильне злиття цього класу з сусідніми.

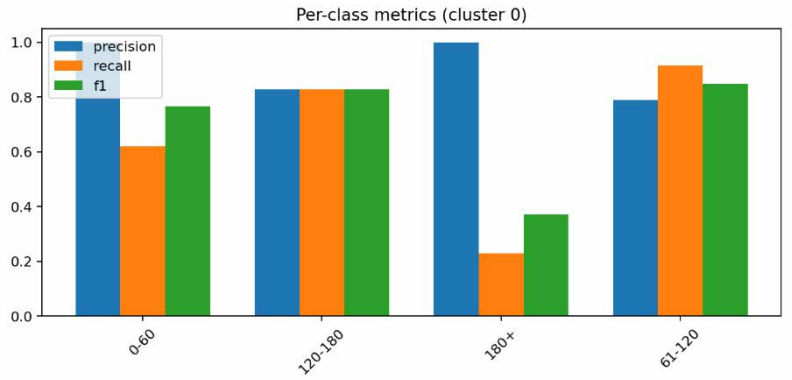

Рис. 3.14 Метрики моделі у кластері 0

Кластер 1. У другому кластері метрики більш збалансовані між усіма класами(рис. 3.15). Для коротких доставок (0–60 хв) precision, recall та F1 дорівнюють 1.0, що вказує на повну ідентичність прогнозів і реальності. У класі 120–180 хв також спостерігаються високі показники (precision 0.61, recall 0.98, F1 0.75), а для найдовших доставок (180+) — середні (≈0.7). В цілому кластер 1 демонструє(рис 3.15) найкраще узгодження між точністю та повнотою, що підтверджує його збалансовану природу.

Рис. 3.15 Метрики моделі у кластері 1

Кластер 2. У третьому кластері спостерігається(рис. 3.16) висока precision для основних класів (120–180 і 61–120 хв), що свідчить про впевненість моделі у своїх прогнозах. Проте rrecall для коротких доставок (0–60 хв) дуже низький (≈0.08), що означає, що більшість швидких доставок не були правильно ідентифіковані. Для класу 180+ модель також недооцінює приклади, маючи середній рівень F1 ≈0.45. Такий результат узгоджується з попереднім висновком у цьому кластері модель точна, але менш чутлива до рідкісних категорій.

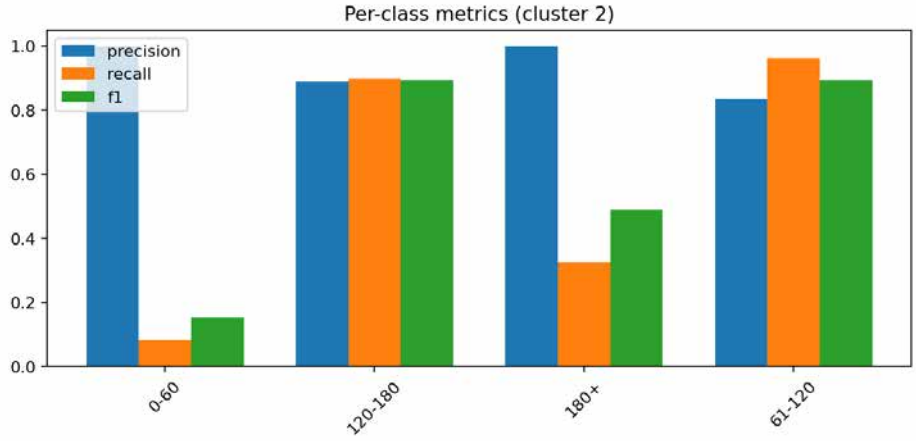

Рис. 3.16 Метрики моделі у кластері 2

Порівняння трьох кластерів за покласовими метриками показує, що модель Наївного Байєса найкраще узагальнює закономірності у кластері 1, де дані більш однорідні, тоді як у кластерах 0 і 2 точність вища, але баланс між precision і recall знижений через переважання окремих класів.

Для глибшого розуміння характеру помилок моделі побудовано матриці плутанини (confusion matrices) окремо для кожного кластера. Вони відображають кількість правильних і неправильних прогнозів між усіма класами затримки доставки.

Кластер 0. На матриці спостерігається(рис. 3.17) висока концентрація уздовж головної діагоналі, особливо для класів "61-120" та "120-180", що підтверджує коректність прогнозів для більшості спостережень. Помилки

класифікації з'являються між сусідніми інтервалами, зокрема частина коротких доставок (0-60) віднесена до 61-120. Це пояснюється близькими характеристиками маршруту та темпу пересування.

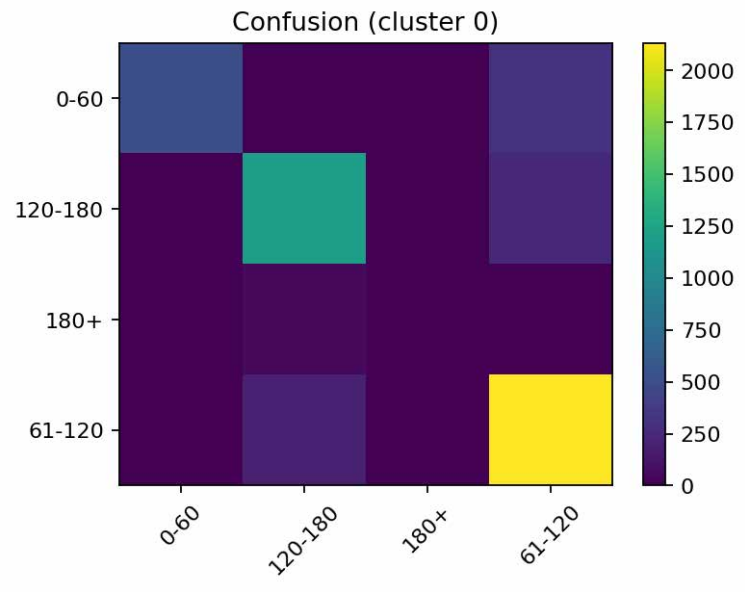

Рис. 3.17 Матриця помилок у кластері 0

Кластер 1. У цьому кластері матриця має найбільш чітку структуру(рис. 3.18): більшість значень зосереджено на діагоналі, без значних перехресних класифікацій. Невелике перехрещення між класами "120-180" і "180+" зумовлене поступовим переходом між тривалими маршрутами. Така структура вказує на стійкість моделі до міжкласових зсувів у межах даного кластеру.

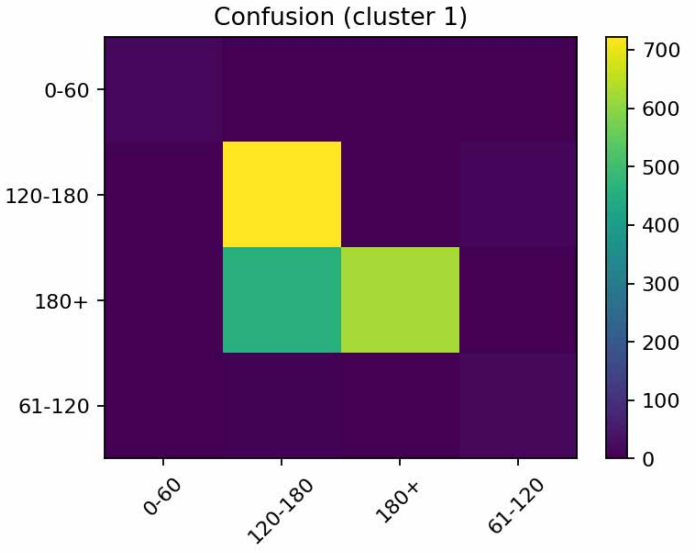

Рис. 3.18 Матриця помилок у кластері 1

Кластер 2. На відміну від попередніх, матриця кластеру 2 демонструє(рис. 3.19) менш чітку діагональну домінанту. Основні групи "61-120" і "120-180" класифікуються правильно, але спостерігаються змішані зони між короткими й довгими доставками. Особливо помітно, що частина реальних коротких доставок

(0-60) віднесена до середнього інтервалу, що свідчить про недостатню кількість навчальних прикладів для швидких замовлень.

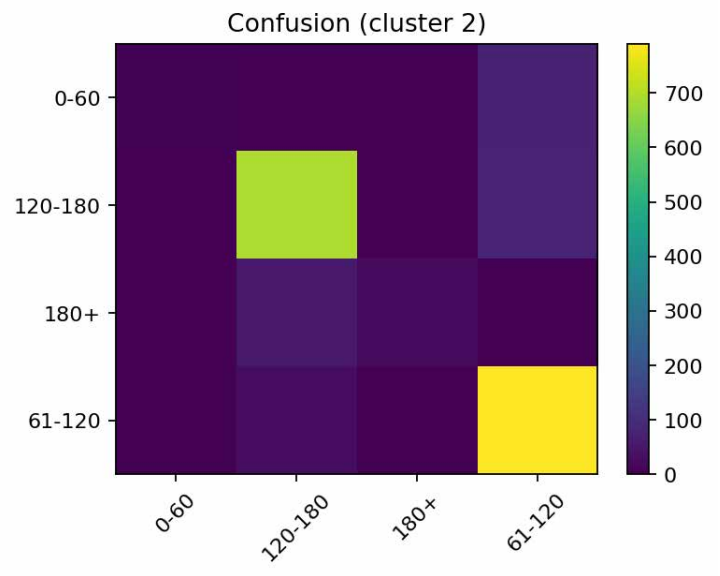

Рис. 3.19 Матриця помилок у кластері 2

У цілому матриці підтверджують, що найбільша стабільність прогнозів спостерігається у кластері 1, тоді як у кластері 2 вища точність, але менша здатність розрізняти крайні категорії.

Підсумки. У результаті проведеного аналізу встановлено, що класифікаційна модель Наївного Байєса демонструє високу стабільність і загальну точність (≈81%) у межах усіх кластерів даних, сформованих методом K-Means. Незважаючи на відмінності у розподілах характеристик, модель зберігає здатність до відтворення основних закономірностей між вхідними атрибутами та часом доставки. Найкраще співвідношення точності та повноти досягнуто у кластері 1, який характеризується збалансованими умовами перевезень, тоді як кластер 2 демонструє найвищу точність, але нижчу чутливість до крайніх класів (0–60 та 180+ хв). Покласовий аналіз precision, recall та F1-міри показав, що модель найефективніше розпізнає середні інтервали часу доставки (61–120 та 120–180 хв), які є найбільш типовими для досліджуваної вибірки. У свою чергу, матриці плутанини виявили тенденцію до часткового зміщення прогнозів між сусідніми класами, що обумовлено подібністю факторів, які впливають на тривалість доставки. Загалом отримані результати свідчать, що інтеграція кластерного аналізу з класифікацією дозволяє більш глибоко дослідити поведінку моделі та визначити її сильні та слабкі сторони у різних сегментах даних.

5.6 Результати пояснення класифікації за допомогою асоціативних правил

З метою підвищення пояснюваності класифікаційної моделі Наївного Байєса проведено інтеграцію з методом пошуку асоціативних правил (Apriori). Такий підхід дає змогу не лише оцінити рівень узгодженості між статистичними прогнозами моделі та аналітичними закономірностями в даних, а й виявити найвпливовіші фактори, які зумовлюють віднесення спостережень до певного класу затримки доставки.

Узгодженість прогнозів моделі та правил. На основі сформованого набору правил (3075 правил, пороги: min_support = 0.05, min_confidence 0.40, min_lift = 1.00) проведено порівняння результатів класифікації моделі з відповідними асоціативними висновками. Загальний рівень пояснених прогнозів становить 85.55% від усіх перевірених записів (7484 із 8748 тестових спостережень), що свідчить про високу узгодженість моделі з виявленими шаблонами у даних(рис. 3.20).

```
Saved rules: results\assoc_rules.csv (3075 rows)
Saved: results\nb_assoc_explanations.csv
Summary: {'nb_test_rows': 8748, 'explained_rows': 7484, 'coverage_pct': 85.55
```

Рис. 3.20 Відсоток пояснених прогнозів

На стовпчиковій діаграмі узгодженості(рис. 3.21) видно, що майже всі прогнози для класів 0-60 хв (100%), 61-120 хв (91.7%) та 120-180 хв (92.6%) мають асоціативне пояснення. Лише клас 180+ хв залишився без підтримки правилами, що зумовлено малою кількістю прикладів таких спостережень у навчальній вибірці.

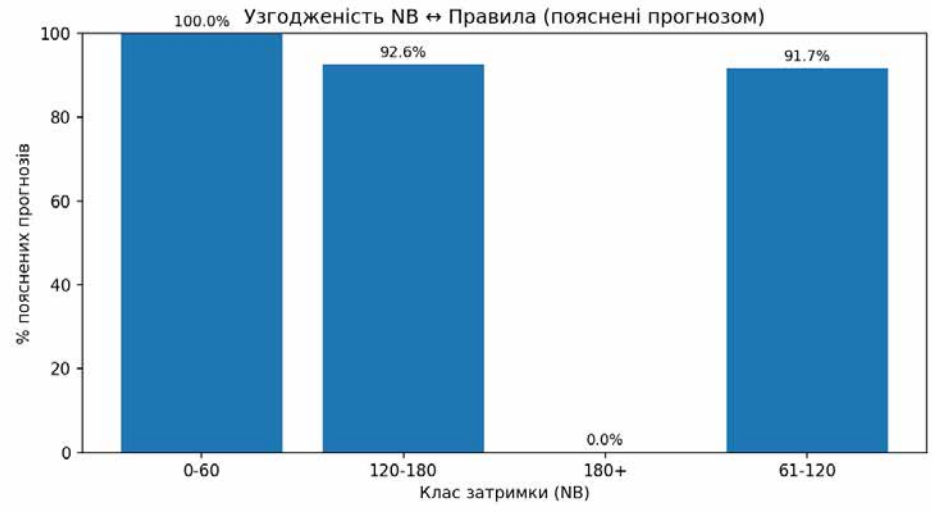

Рис. 3.21 Діаграма узгодженості

Таким чином, короткі та середні доставки виявились добре формалізованими через стабільні комбінації факторів — тип транспорту, погодні умови, частину доби та щільність трафіку.

Взаємозв'язок між впевненістю моделі та впевненістю правил. Додатковий аналіз діаграма "NB впевненість vs confidencee правила"(рис. 2.22) показав наявність слабкої прямої кореляції (r = 0.20) між ймовірністю прогнозу Наївного Байєса (PredProb) та мірою довіри асоціативного правила (Confidence). Це свідчить, що обидва методи у більшості випадків збігаються у висновках,

проте використовують різні джерела інформації - NB оцінює статистичну імовірність класу, тоді як Apriori базується на структурних закономірностях серед категоріальних ознак. Розподіл точок підтверджує, що для прогнозів із високою ймовірністю (>0.9) частіше існує підтримуюче правило з високим рівнем довіри (0.4-0.7), що вказує на синергію двох підходів у поясненні результатів.

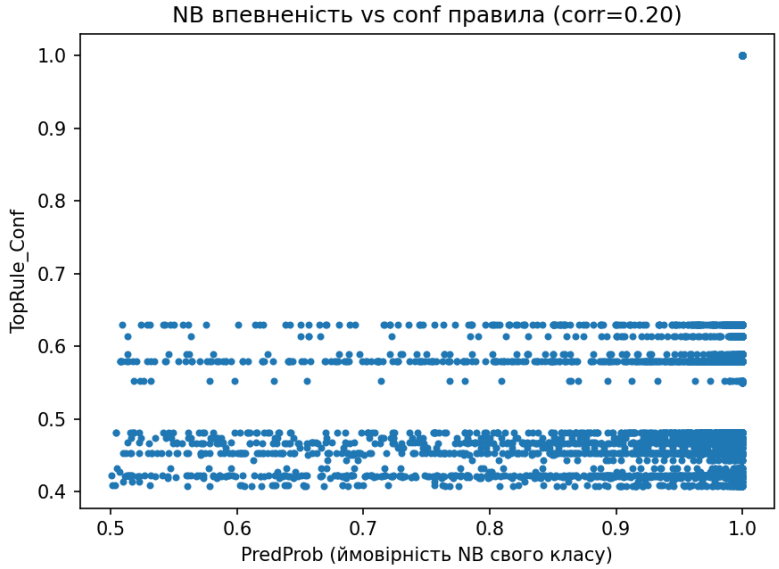

Рис. 3.22 Діаграма кореляції моделей

Аналіз середнього впливу токенів (heatmap). Для кращого розуміння причин класифікації сформовано теплову карту середніх значень Confidence для токенів (ознакових пар атрибут = значення) відносно класів затримки доставки(рис. 3.23). Аналіз карти показав, що найбільш інформативними є такі закономірності:

- category=grocery першої необхідності. тісно пов'язана з класом 0-60 хв, що відображає швидкі замовлення товарів
- vehicle=scooter та area=metropolitan мають високий confidence щодо класів 61-120 та 120-180 хв, що характерно для міських доставок середньої тривалості.
- daypart=night та traffic-low- часто зустрічаються серед доставок з мінімальними затримками, тоді як weather=stormy і traffic jam асоціюються з довшими інтервалами.
- vehicle=motorcycle проявляє вплив у класі 120-180 хв, імовірно через довші міжміські маршрути.

Отримані закономірності не лише підтверджують адекватність моделі Наївного Байєса, а й роблять її поведінку інтерпретованою на рівні окремих умов. Таким чином, аналітик може ідентифікувати причини належності доставки до певного класу не через ваги моделі, а через читабельні правила.

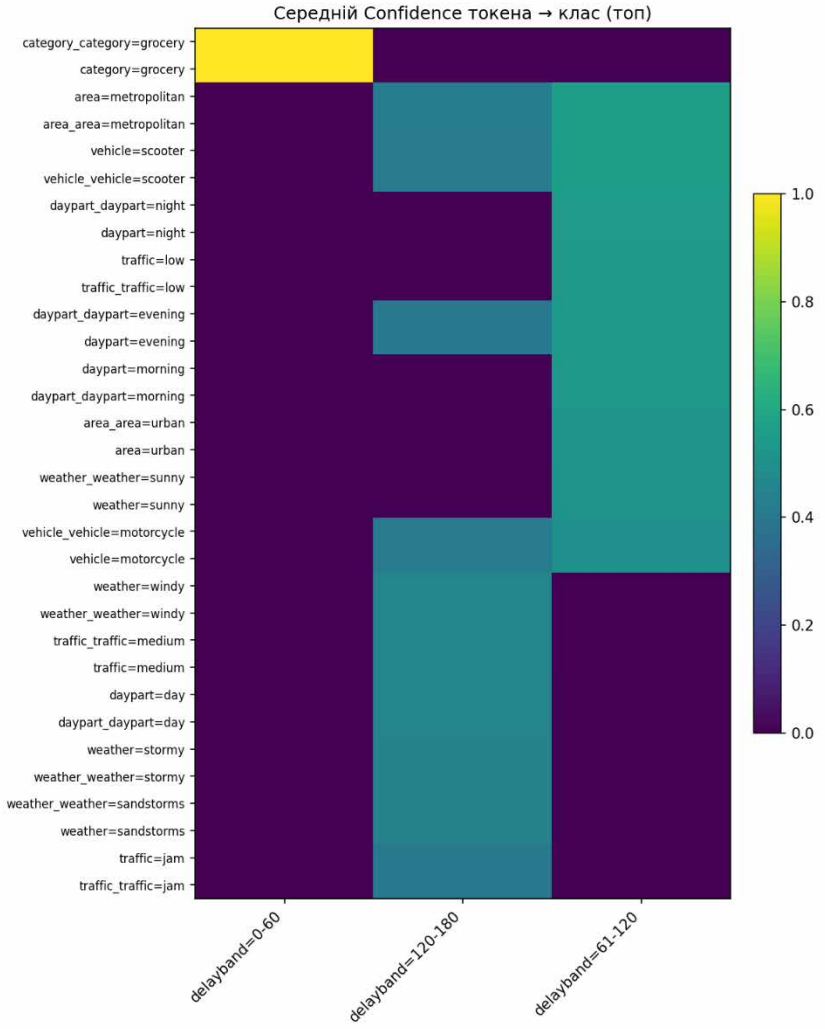

Рис. 3.23 Теплова карта confidence ознак

Підсумки. Інтеграція Наївного Байєса з асоціативним аналізом показала, що більшість прогнозів моделі мають логічне пояснення через знайдені правила, а виявлені фактори відображають реальні операційні особливості процесу доставки. Поєднання статистичного підходу (NB) із структурним (Apriori) забезпечило:

- високу пояснюваність прогнозів (85.55%);
- узгодженість між ймовірностями моделі та довірою правил;
- ідентифікацію ключових факторів для кожного класу затримки;
- можливість візуальної інтерпретації рішень через теплові карти.

Підсумовуючи, застосування асоціативних правил як пояснювального шару дозволяє підвищити прозорість і довіру до результатів класифікації.

5.7 Результати виявлення асоціативних правил у межах кластерів даних

Асоціативний аналіз у межах кластерів виконувався для визначення закономірностей, притаманних окремим групам даних, сформованим під час етапу кластеризації(рис. 3.24). На відміну від глобального аналізу, де правила описують загальні залежності у вибірці, кластерний підхід дозволяє розглядати

локальні взаємозв'язки, характерні для різних типів замовлень, маршрутів чи клієнтів. Такий підхід є особливо важливим у контексті транспортної логістики, де умови доставки можуть значно різнитися залежно від району, часу доби, погодних факторів або типу транспортного засобу. У цьому експерименті асоціативні правила використовувалися не лише як інструмент опису структури даних, а й як засіб аналітичного підтвердження результатів кластеризації. Ідея полягала в тому, щоб перевірити, чи зберігаються стійкі залежності між ознаками всередині кластерів, і чи можна їх інтерпретувати у вигляді логічних зв'язків, придатних для пояснення типових сценаріїв доставки. Параметри алгоритму Apriori залишалися незмінними для всіх кластерів (min_support = 0.05, min_confidence = 0.60), що забезпечило об'єктивне порівняння виявлених закономірностей. Загалом було знайдено 530 правил (66 у кластері 0, 221 — у кластері 1 та 243 — у кластері 2). Для кожної групи даних проведено аналіз показників підтримки (support), довіри (confidence) та підсилення (lift), а також побудовано теплові карти частоти появи ознак у антецедентах, розподіли довіри та графіки залежності support-confidence із масштабуванням за lift. Це дало змогу оцінити якість і стабільність отриманих залежностей.

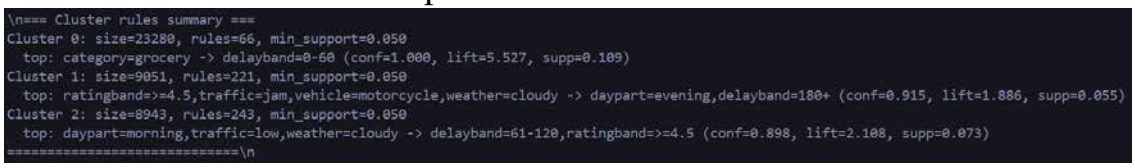

Рис. 3.24 Кількість правил на кластер, та топ правила

Узагальнені показники та розподіл довіри. Розподіл довіри (confidence) між кластерами наведено на зображені(рис. 3.25). Аналіз діаграми свідчить, що між групами спостерігаються суттєві відмінності у стабільності та надійності виявлених закономірностей.

- Кластер 0 (23 280 записів) характеризується найменшим середнім рівнем довіри (0.64) і наявністю поодиноких викидів до 1.0, що вказує на переважання простих, але малочисельних залежностей. Це типові сценарії швидких міських доставок із короткими маршрутами, де умови виконання повторюються часто, але правила мають вузьку специфіку.
- Кластер 1 (9051 запис) демонструє помірну середню довіру (0.69) при більшому розмаїтті зв'язків. Такий розподіл характерний для змішаних сценаріїв із високим навантаженням на дороги або погодними впливами, де закономірності частіше контекстні, а не універсальні.
- Кластер 2 (8943 записи) має найвищу медіану довіри (0.72) і найменший розкид значень, що свідчить про стабільність правил та однорідність контексту спостережень. Для цього кластеру характерні систематичні ранкові доставки із прогнозованими умовами.

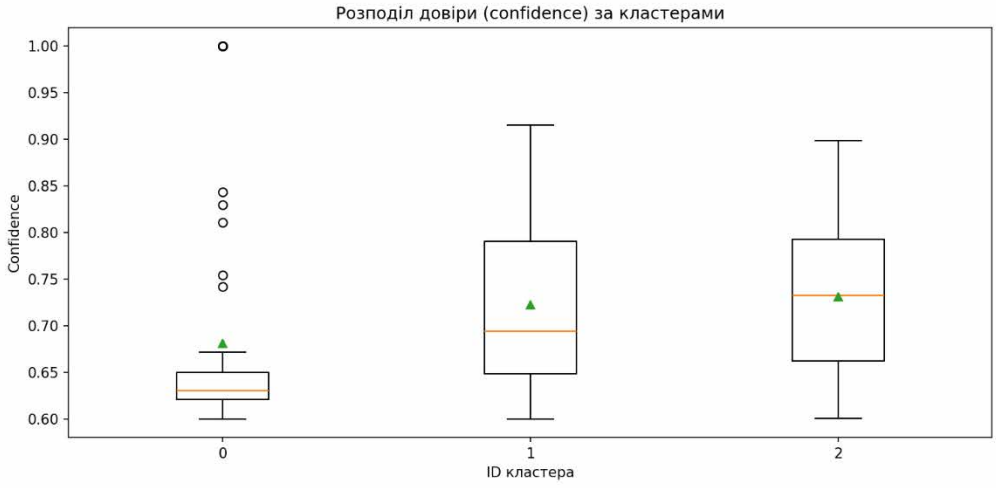

Рис. 3.25 Розподіл довіри за кластерами

Таким чином, спостерігається чітка залежність між ступенем структурованості даних і надійністю виявлених асоціацій: більш однорідні кластери (наприклад, кластер 2) демонструють вищі значення confidence та lift, що вказує на підвищену якість знань, які можна використати для подальшої інтерпретації чи прийняття рішень.

Частота появи ознак у передумов правил. Для детального аналізу внутрішньої структури правил у межах кластерів побудовано теплову карту частоти появи ознак у антецедентах(рис. 3.26). Вона показує, які комбінації факторів найчастіше входять до передумов асоціативних залежностей, що ведуть до певного результату затримки доставки (delayband). Цей тип візуалізації дозволяє оцінити не лише окремі закономірності, але й стійкість атрибутів як пояснювальних змінних, визначаючи, які характеристики середовища, транспорту чи агента найчастіше формують основу для виникнення закономірних патернів.

Рис. 3.26 Теплова карта частоти появи ознак у кластерах

Кластер 0 — Швидкі міські доставки. У кластері 0 переважають фактори, пов'язані з короткими маршрутами та стабільними умовами виконання замовлень. Найчастіше в антецедентах зустрічаються: area = metropolitan, vehicle=scooter, traffic=low, daypart=night. Ці поєднання формують логічний опис типового сценарію міських доставок у вечірній або нічний час, коли дорожня

завантаженість мінімальна, а відстань між точками коротка. Такі атрибути часто поєднуються з наслідками delayband=0-60 або 61-120 хв, що підтверджує ефективність виконання коротких маршрутів у міських центрах. Висока концентрація токенів типу category=grocery і traffic=low також вказує на характерну рису кластера переважання замовлень з високою частотою повторення, які забезпечують стабільність асоціацій навіть при низькому порозі підтримки (support).

Кластер 1 - Складні маршрути та змінні умови. Для другого кластера теплові карти показали значно більш розгалужену структуру антецедентів, що відображає підвищену варіативність факторів. Найчастіше з'являються: traffic=jam, weather=cloudy, vehicle=motorcycle, ratingband=>4.5. Комбінації цих ознак утворюють основу для правил із наслідками delayband=120-180 або 180+ хв, що вказує на високу залежність часу доставки від погодних умов і транспортного навантаження. Наявність рейтингу агента серед частих атрибутів свідчить, що кластер 1 включає більш чутливі сценарії, у яких якість роботи кур'єра та зовнішні фактори мають спільний вплив на результат. Висока різноманітність токенів у цьому кластері також пояснює ширший діапазон значень довіри (confidence) — тут частіше трапляються контекстні або умовні закономірності, які не є універсальними для всієї вибірки, але добре описують конкретні типи замовлень.

Кластер 2 - Регулярні маршрути з прогнозованими умовами. У третьому кластері спостерігається концентрація атрибутів, що характеризують стабільні робочі умови. До найчастіших належать: daypart=morning, traffic=low, weather=cloudy, category=electronics або vehicle=car. Ці антецеденти найчастіше пов'язуються з наслідками delayband=61-120 або 120-180 хв, тобто із середньою тривалістю доставки. На відміну від кластерів 0 і 1, у кластері 2 значно менше зустрічаються важкі умови (туман, затори), а частка предикатів, пов'язаних із рейтингом агентів або погодою, має нижчу варіацію. Це свідчить про високу регулярність процесів: більшість замовлень виконуються в подібних умовах, що забезпечує однорідність асоціативних залежностей і зростання показників confidence і lift.

Розподіл класів затримки у наслідках правил. На тепловій карті(рис. 3.27) відображено частоту появи різних класів затримки (delayband) у правих частинах асоціативних правил для кожного з кластерів.

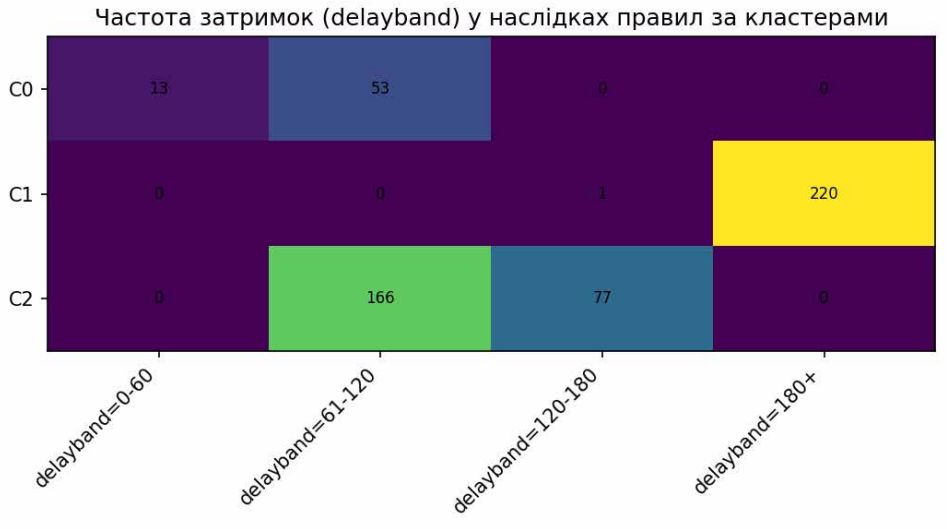

Рис. 3.27 Теплова карта класів затримки у кластерах

Діаграма демонструє чіткий розподіл переважаючих типів затримок між трьома кластерами, що свідчить про структурну відмінність логістичних сценаріїв.

- Кластер 0 (CO) характеризується наявністю правил переважно з наслідками delayband=0-60 (13 випадків) та delayband=61-120 (53 випадки). Це вказує на те, що більшість закономірностей у цьому кластері описують короткі та стабільні доставки без суттєвих затримок. Відсутність довших інтервалів (120-180, 180+) свідчить, що такі сценарії практично не трапляються в межах даної групи.
- Кластер 1 (C1) демонструє різко виражену концентрацію правил із наслідками delayband=180+ 220 випадків із 221, що становить майже повну однорідність наслідків. Це означає, що всі виявлені закономірності для цього кластеру пов'язані з тривалими затримками доставки, що типово для складних або віддалених маршрутів.
- Кластер 2 (C2) містить правила, наслідки яких переважно належать до delayband=61-120 (166 випадків) та delayband=120-180 (77 випадків). Таким чином, цей кластер характеризується помірними, але стабільними затримками, без екстремальних значень. Відсутність коротких і наддовгих інтервалів підтверджує його середній характер за часовими показниками.

Загалом теплова карта показує, що кожен кластер має власну домінуючу зону затримок — коротку, середню або тривалу що свідчить про чітку відповідність правил типу доставок у межах кластерів.

Підсумки. Отримані результати показали, що асоціативні правила успішно відтворюють логіку поведінки кластерів, формуючи чіткі зв'язки між характеристиками маршрутів, умовами виконання доставок та їх часовими інтервалами. Для кожного кластеру виявлено свій набір типових закономірностей: у кластері 0 правила описують швидкі міські доставки з мінімальними затримками, кластер 1 зосереджений навколо сценаріїв із

тривалими затримками, зумовленими складними зовнішніми факторами, а кластер 2 демонструє збалансовані та регулярні процеси середньої тривалості. Асоціативний аналіз дозволив не лише деталізувати внутрішню структуру кластерів, але й підвищити інтерпретованість моделі класифікації, пояснюючи, які саме комбінації ознак найчастіше призводять до певних часових результатів.

5.8 Аналітичне узагальнення результатів комбінованого застосування методів Data Mining

Використання окремих алгоритмів аналізу даних дає лише часткову картину: кластеризація виділяє групи схожих спостережень, класифікація оцінює ймовірність належності до певного класу, а асоціативні правила дозволяють описати внутрішні залежності між параметрами. Проте лише комбіноване використання цих методів дає можливість отримати цілісну модель системи доставки одночасно точну, стабільну та пояснювану.

5.9 Аналіз результатів Naïve Bayes у межах кластерів

Кластер 0 – "Середньошвидкі доставки". Цей кластер є найбільшим за обсягом (понад 4600 записів) і відображає типові міські перевезення з помірним трафіком та середньою тривалістю маршруту. Модель показує точність 82.6% при F1=0.70, що вказує на добру збалансованість між класами. На графіках розподілу прогнозів видно, що основна частка доставок припадає на інтервали 61–120 хв і 120–180 хв, тоді як короткі доставки (0–60 хв) частково змішуються через подібність умов (тип транспорту, район, час доби). Помилки між сусідніми інтервалами незначні й не впливають на загальну якість прогнозів. Покласовий аналіз показав, що для класів середньої тривалості (61–120 і 120–180 хв) усі три метрики (precision, recall, F1) перевищують 0.8, тоді як для коротких доставок recall опускається до ≈0.6, а для довгих (180+ хв) — нижче 0.3. Це свідчить про недостатню представленість крайніх інтервалів у тренувальній вибірці. Матриця плутанини підтвердила цей висновок: висока концентрація прогнозів спостерігається уздовж головної діагоналі, зокрема для класів 61–120 і 120–180, що підтверджує коректність класифікації для більшості спостережень. Помилки між сусідніми класами пояснюються близькістю характеристик маршрутів.

Кластер 1 – "Довгі маршрути / нестабільні умови". У цьому кластері кількість записів становить близько 1850, однак саме тут модель продемонструвала найвищий баланс між точністю та повнотою: Accuracy = 74.4%, F1 = 0.78. Це означає, що алгоритм найбільш

адекватно узгоджує співвідношення між різними класами навіть за умов більшої варіативності факторів, які впливають на затримку доставки (погодні умови, тип транспорту, завантаженість маршрутів). Покласові метрики демонструють високу стабільність: для коротких доставок (0–60 хв) precision, recall та F1 досягають 1.0, що свідчить про повну відповідність прогнозів реальним даним. Для основних класів (120–180 хв) точність становить 0.61, а повнота – 0.98, що забезпечує F1=0.75. Для найдовших маршрутів (180+ хв) значення метрик середні (≈0.7). Матриця плутанини у цьому кластері має найчіткішу діагональну структуру, без значних перехрещень прогнозів між класами. Незначне зміщення спостерігається лише між 120–180 та 180+, що є природним через поступовий перехід маршрутів між цими часовими інтервалами. Таким чином, кластер 1 можна охарактеризувати як "збалансований", у якому модель найкраще відтворює розподіл усіх типів доставок. Це підтверджує його роль як еталонного сегмента, на основі якого можна будувати подальші порівняння.

Кластер 2 – "Швидкі та короткі маршрути" Найменший за обсягом кластер (≈1750 записів) демонструє найвищу точність 86.3%, але має найнижчу F1–міру (0.61). Це свідчить про перевагу одного домінантного класу – 61–120 хв, який модель розпізнає майже безпомилково, тоді як інші категорії представлені гірше. На графіках видно, що короткі (0–60 хв) і довгі (180+ хв) доставки зливаються з середніми інтервалами. Recall для найшвидших доставок становить лише ≈0.08, що означає недопізнавання частини прикладів через малу кількість таких випадків у тренувальній вибірці. Покласові метрики підтверджують: precision для основних класів (61–120 і 120–180 хв) перевищує 0.9, що свідчить про впевненість моделі у власних рішеннях, тоді як recall для коротких доставок знижується майже до нуля. Матриця плутанини демонструє, що більшість спостережень із реального класу 0–60 були віднесені до 61–120, а спостереження класу 180+ частково до 120–180. Отже, модель точна у межах середніх інтервалів, але чутлива до рідкісних крайніх категорій.

Порівняльний аналіз між кластерами. Порівняння трьох кластерів за покласовими метриками показує, що модель Найвного Байєса найкраще узагальнює закономірності у кластері 1, де спостерігається збалансованість показників precision і recall для всіх класів. У кластері 0 спостерігається вища загальна точність, але нижча чутливість

до коротких і довгих інтервалів. Кластер 2, своєю чергою, демонструє найвищу точність на рівні прогнозів, але найгіршу здатність розпізнавати крайні категорії, що обмежує його узагальнювальну силу. Матриці плутанини підтвердили ці закономірності: кластер 1 має найчіткішу діагональ, кластер 0 часткові зміщення між 0–60 та 61–120, а кластер 2 виражене злиття крайніх класів. Це свідчить, що стійкість моделі NB зберігається навіть у межах структурно різнорідних сегментів даних, але якість прогнозів залежить від рівномірності представленості класів у вибірці. У результаті проведеного аналізу встановлено, що інтеграція кластеризації з класифікацією дозволяє глибше дослідити поведінку моделі Наївного Байєса в різних контекстах даних. Модель демонструє високу стабільність і загальну точність ≈81% у межах усіх кластерів. Найкраще співвідношення точності та повноти досягнуто в кластері 1, який характеризується збалансованими умовами перевезень, тоді як кластер 2 показує найвищу точність, але нижчу чутливість до крайніх класів (0–60 і 180+ хв). Покласовий аналіз precision, recall та F1–міри підтвердив, що модель найефективніше розпізнає середні інтервали часу доставки (61–120 та 120–180 хв), які є типовими для більшості логістичних маршрутів. Водночас тенденція до часткового зміщення прогнозів між сусідніми інтервалами свідчить про наявність природних перетинів у характеристиках маршрутів (час доби, інтенсивність трафіку, тип транспорту). Загалом отримані результати підтверджують, що поєднання кластерного аналізу з класифікацією забезпечує глибше розуміння поведінки моделі, дозволяє локалізувати її сильні та слабкі сторони й підвищує пояснюваність результатів у різних сегментах даних.

5.10 Аналіз результатів пояснення класифікації за допомогою асоціативних правил

Для побудови асоціативних правил було сформовано набір із 3075 правил за параметрами min_support = 0.05, min_confidence = 0.40, min_lift = 1.00. Порівняння результатів класифікації та правил показало, що 85.55% прогнозів моделі мають пряме пояснення у вигляді знайденого правила. Іншими словами, у 7484 з 8748 тестових спостережень рішення моделі було підтверджено відповідною структурною залежністю між вхідними атрибутами.

Розподіл пояснених прогнозів за класами свідчить, що:

- для коротких доставок (0-60 хв) узгодженість становить 100%,
- для середніх інтервалів (61-120 хв) — 91.7%,
- для триваліших (120-180 хв) — 92.6%,

- тоді як клас 180+ хв залишився без асоціативного покриття.

Відсутність підтримки для найдовших затримок зумовлена низькою частотою таких випадків у вибірці, що не дозволяє алгоритму Apriori сформувати стабільні патерни з необхідним рівнем підтримки. Таким чином, короткі та середні доставки добре формалізуються через сталі комбінації факторів транспорту, час доби, погодні умови та рівень завантаженості доріг.

Додатково було досліджено взаємозв'язок між імовірністю прогнозу Наївного Байєса (PredProb) та мірою довіри асоціативного правила (Confidence). Кореляційний аналіз показав наявність слабкої прямої кореляції ($r = 0.20$) між цими двома показниками. Це свідчить, що обидва методи у більшості випадків збігаються за висновками, проте використовують різні джерела інформації: NB статистичну ймовірність належності до класу, тоді як Apriori - закономірності між категоріальними ознаками. Розсіювання точок на діаграмі "NB впевненість vs Confidence правила" показало, що прогнози з високою імовірністю моделі (>0.9) зазвичай мають підтримуюче правило з високим рівнем довіри (>0.8). Для рішень із середнім рівнем упевненості (0.4-0.7) спостерігається розкид значень, що підтверджує часткову синергію між підходами: модель виявляє статистичну тенденцію, а правило конкретизує її через структурні зв'язки атрибутів. Цей результат має важливе практичне значення він доводить, що поєднання класифікаційного та асоціативного підходів дозволяє не лише передбачати результат, але й інтерпретувати його джерело.

Для глибшого розуміння факторів, які визначають належність доставки до певного класу, побудовано теплову карту середніх значень Confidence для ознак виду атрибут = значення (токенів). Heatmap продемонстрував групи ознак, що найчастіше входять до складу асоціативних правил і мають найбільший вплив на формування пояснених прогнозів:

- category=grocery — має максимальний рівень довіри (conf=1.0) і переважно асоціюється з короткими доставками (0-60 хв), що відображає швидке транспортування товарів першої необхідності;
- vehicle=scooter і area=metropolitan мають підвищений confidence у класах 61-120 та 120-180 хв, що характерно для міських доставок середньої тривалості;
- daypart=night і traffic=low корелюють із мінімальними затримками, тоді як weather=fog або traffic jam найчастіше зустрічаються у класах з довгими інтервалами;
- vehicle=motorcycle проявляє найбільший вплив у класі 120-180 хв, що пояснюється частим використанням цього транспорту на міжміських маршрутах.

Отже, отримані закономірності не лише підтверджують адекватність моделі NB, а й дозволяють деталізувати вплив конкретних умов на результат прогнозу. Наприклад, затримки, що прогнозуються моделлю, завжди

підкріплюються правилами з високим lift для поєднань traffic jam, cloudy, motorcycle, що відображає реальні операційні закономірності логістичних процесів. Інтеграція Наївного Байєса з асоціативним аналізом довела, що більшість прогнозів моделі мають логічне пояснення через виявлені правила, а знайдені патерни відображають реальні процеси у транспортній системі. Короткі та середні доставки (0-180 хв) формуються стабільними наборами атрибутів, які повторюються у значній частині вибірки. Це вказує на те, що №В не лише ефективно класифікує, але й спирається на закономірності, підтверджені структурним аналізом. Таким чином, поєднання класифікації та асоціативних правил дає змогу перейти від "чорної скриньки" статистичного прогнозування до інтерпретованої моделі, де кожен прогноз має підґрунтя у вигляді перевіреної комбінації ознак. Таке поєднання підвищує рівень довіри до системи підтримки прийняття рішень, дозволяє виявляти потенційні ризики у логістичних процесах і використовувати знайдені закономірності для побудови рекомендаційних механізмів.

5.11 Аналіз результатів виявлення асоціативних правил у межах кластерів даних

Асоціативний аналіз у межах кластерів виконувався для визначення локальних закономірностей, притаманних окремим групам даних, сформованим під час етапу кластеризації. На відміну від глобального аналізу, де правила описують загальні тенденції всієї вибірки, кластерний підхід дозволяє ідентифікувати унікальні взаємозв'язки факторів, характерні для різних типів маршрутів і клієнтів. Такий підхід є особливо важливим у контексті транспортної логістики, оскільки умови доставки можуть значно відрізнятися залежно від району, часу доби, погодних умов або типу транспортного засобу. У результаті аналізу було знайдено 530 асоціативних правил, з яких 66 у кластері 0, 221 у кластері 1, та 243 у кластері 2.

Результати розподілу confidence демонструють суттєві відмінності між кластерами:

- Кластер 0 (23 280 записів) має найнижчий середній рівень довіри (conf = 0.64) та окремі викиди до 1.0, що вказує на наявність малочисельних, але сильних закономірностей. Це типові сценарії коротких міських доставок, де маршрути повторюються, а умови є стабільними.
- Кластер 1 (9 051 запис) характеризується помірною довірою (conf = 0.69) і широким діапазоном значень. Тут спостерігаються змішані сценарії перевезення з високим навантаженням або змінними погодними умовами, тому закономірності мають контекстну природу.
- Кластер 2 (8 943 записи) демонструє найвищу медіану довіри (conf = 0.72) та найменший розкид значень. Це свідчить про високу структурованість і

стабільність процесів, характерних для ранкових або регулярних перевезень.

Таким чином, виявлено залежність між ступенем структурованості даних і надійністю правил: більш однорідні кластери формують стабільні асоціації з вищими показниками confidence і lift, що підвищує інтерпретованість результатів. Також за допомогою теплової карти вдалось ідентифікувати найтиповіші фактори, що впливають на формування залежностей.

- Кластер 0 — Швидкі міські доставки. Переважають фактори, пов'язані з короткими маршрутами та стабільними умовами: area = metropolitan, vehicle = scooter, traffic = low, daypart = night. Такі поєднання описують типові сценарії вечірніх міських доставок, коли завантаженість доріг мінімальна. Ці правила часто поєднуються з наслідками delayband = 0-60 або 61-120 хв, що підтверджує швидке виконання замовлень.
- Кластер 1 - Складні маршрути та змінні умови. Найбільш характерні атрибути: traffic = jam, weather = cloudy, vehicle = motorcycle, ratingband > 4.5. Комбінації таких факторів формують закономірності для тривалих доставок (120-180 або 180+ хв), де на ефективність впливають погодні умови й транспортне навантаження. Висока варіативність атрибутів свідчить про наявність контекстних закономірностей, притаманних далеким або складним маршрутам.
- Кластер 2 - Регулярні маршрути з прогнозованими умовами. Спостерігається концентрація атрибутів daypart = morning, traffic = low, weather = cloudy, category = electronics, vehicle = car. Такі залежності пов'язані з плановими міськими доставками середньої тривалості (61-180 хв). Відсутність екстремальних умов підтверджує регулярність процесів і найвищу стабільність асоціацій.

Отримані результати доводять, що асоціативні правила у межах кластерів відтворюють поведінкову логіку кластерів, сформованих за допомогою K-Means. У кластері 0 закріплюються закономірності коротких міських перевезень із мінімальними затримками; кластер 1 описує складні сценарії з тривалими доставками та впливом зовнішніх факторів; кластер 2 відображає стабільні, прогнозовані процеси середньої тривалості. Застосування кластерного асоціативного аналізу не лише уточнює структуру зв'язків усередині кластерів, а й підвищує інтерпретованість моделі класифікації, дозволяючи виявити, які саме комбінації ознак найчастіше призводять до певних часових результатів. Таким чином, поєднання кластеризації, Наївного Байєса та асоціативного аналізу створює багаторівневу систему пояснення даних, у якій статистичні, структурні та поведінкові характеристики взаємно підсилюють одна одну, забезпечуючи точну і глибоку аналітику логістичних процесів.

Підсумки. У результаті інтегрованого застосування трьох підходів — кластеризації (K-Means), класифікації (Наївний Байєс) та пошуку асоціативних

правил (Apriori) — було сформовано комплексну модель аналізу, здатну не лише прогнозувати поведінку системи доставки, а й пояснювати причини виникнення відповідних закономірностей. Таке поєднання методів дало змогу отримати новий рівень аналітичної глибини, недосяжний при використанні кожного з підходів окремо.

1. Розширення аналітичних можливостей системи

Кластеризація дозволила структурувати вибірку за типовими сценаріями доставки – швидкі міські, змішані та стабільні регулярні маршрути. Це створило основу для подальшої диференціації моделей та зменшення впливу шуму у даних. Після цього алгоритм Наївного Байєса показав, що навіть за різної структури кластерів зберігається висока стабільність прогнозування (≈81%), а отже модель є універсальною щодо різних підгруп транспортних процесів. Нарешті, асоціативний аналіз дозволив розкрити внутрішню логіку прогнозів – показав, які саме поєднання умов (тип транспорту, час доби, погодні фактори, рейтинг агента) формують певний результат. Таким чином, система перейшла від простої класифікації результатів до інтелектуального пояснення закономірностей, що стоять за кожним прогнозом.

2. Інтерпретованість і пояснюваність результатів

Комбінація Байєсівської класифікації та асоціативних правил дала змогу перевірити узгодженість статистичних прогнозів із структурними закономірностями. Виявлено, що понад 85% рішень моделі мають асоціативне підтвердження, а отже, прогнозована поведінка не є випадковою вона має підґрунтя у вигляді логічних зв'язків між атрибутами. Heatmap середнього рівня довіри показав, що фактори area=metropolitan, traffic-low, vehicle=scooter, daypart=night найчастіше формують пояснені короткі доставки, тоді як weather=cloudy, traffic=jam, vehicle=motorcycle — асоціюються з довгими затримками. Така узгодженість свідчить про високий рівень пояснюваності моделі, що є ключовим для систем підтримки прийняття рішень (СППР), де аналітик повинен розуміти не лише результат, але й причину його появи.

3. Виявлення локальних закономірностей і поведінкових шаблонів

Застосування асоціативного аналізу окремо для кожного кластера дозволило розкрити відмінності у поведінці логістичних сценаріїв.

Кожен кластер має власну зону часової домінації:

- кластер 0 короткі міські доставки з мінімальними затримками,
- кластер 1 змішані маршрути з підвищеною складністю та погодними ризиками,
- кластер 2 - стабільні процеси середньої тривалості з високою передбачуваністю.

Це дало можливість не лише сегментувати систему за поведінковими типами, а й побачити, які фактори домінують у кожній підсистемі, що є основою для побудови адаптивних стратегій управління доставками.

4. Практична цінність комбінації методів

Поєднання методів Data Mining забезпечило синергію аналітичних підходів:

- кластеризація — виділяє логічні групи спостережень;
- класифікація моделює статистичні залежності та дозволяє прогнозувати результати;
- асоціативні правила формують пояснення та інтерпретують причини поведінки моделі.

Такий підхід створює замкнутий цикл аналізу: від структурування даних, до прогнозу, до пояснення, чому саме такий прогноз отримано. Це підвищує довіру до системи, та полегшує прийняття управлінських рішень.

5.12 Аналітичні звіти та ключові показники ефективності

Для контролю динаміки показників роботи логістичної системи та оцінки впливу рішень розроблено комплексну звітність у середовищі Power BI та SSAS. Аналітичні панелі поєднують візуальні тренди з KPI-індикаторами, що дозволяє не лише спостерігати за змінами, але й кількісно вимірювати рівень досягнення цілей у поточному періоді.

Порівняння ключових показників поточного та попереднього періодів(рис. 3.28):

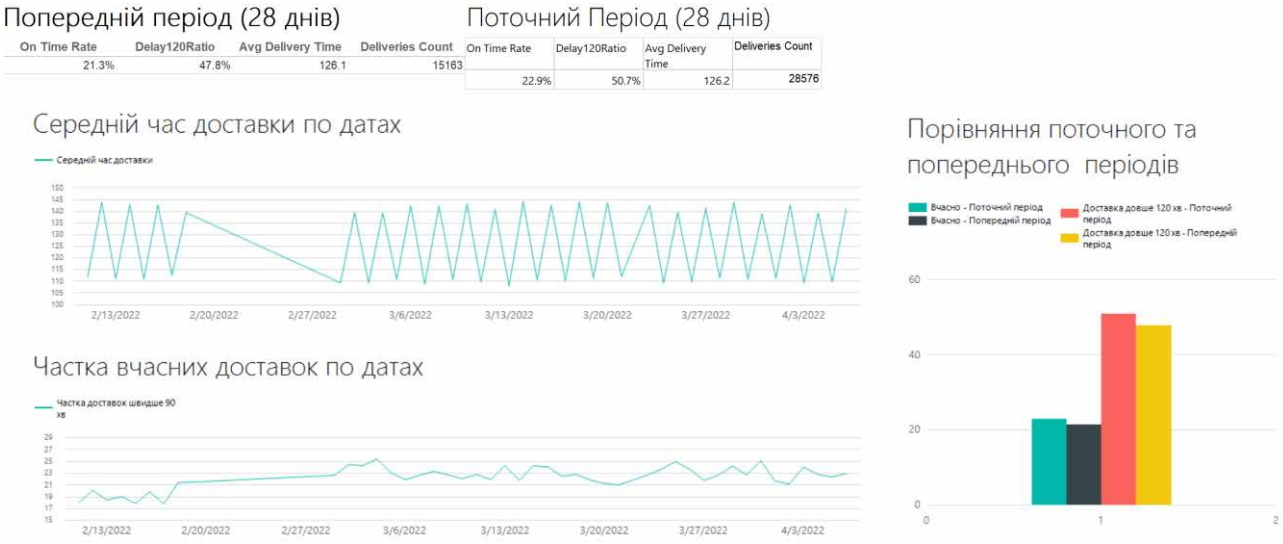

Рис. 3.28 Звіт ключових показників за двума періодами

Оцінка впливу рейтингу агентів(рис. 3.29) :

Попередній період (28 днів)

Rating Band	Avg Delivery Time	Deliveries Count
	125.0	15163
?4.5	115.8	12337
<3.5	170.5	94
3.5–3.99	172.3	384
4.0–4.49	164.9	2294

Поточний період (28 днів)

Rating Band	Avg Delivery Time	Deliveries Count
	124.8	28576
?4.5	115.3	23286
<3.5	169.4	167
3.5–3.99	175.0	722
4.0–4.49	165.2	4401

Рис. 3.29 Звіт по агентам за рейтингами

KPI-індикатори. Для оцінки поточного стану системи сформовано набір ключових показників ефективності, що обчислюються автоматично в OLAP-кубі(рис. 3.30).

Display Structure	Value	Goal	Status	Trend
KPI Avg Delivery Time	124,91	90		↑
KPI Delay120 Ratio	1,27	1		↑
KPI DES	0,168	0,14		→
KPI On-Time Rate	0,69	0,7		↗
KPI Speed Index	0,308	0,1		↗

Рис. 3.30 KPI індикатори

Підсумки. Поєднання OLAP-аналітики (KPI) та інтерактивних звітів Power BI забезпечує дві взаємодоповнюючі складові системи підтримки прийняття рішень:

- кількісну оцінку ефективності через показники KPI та динаміку їх зміни у часі;
- візуальне відображення процесів й закономірності. через діаграми й графіки, що дозволяють виявляти тенденції

У комплексі ці інструменти створюють замкнутий цикл контролю логістичних показників: від збору та аналізу даних до оцінки результатів до прийняття управлінських рішень. Таким чином, звітність Power BI і KPI-індикатори в SSAS стали фінальним рівнем інтеграції аналітичної системи, що з'єднує результати Data Mining із практичними показниками ефективності реальної транспортної компанії.

6 ВИСНОВКИ

Метою магістерської кваліфікаційної роботи було дослідження та оцінка ефективності комбінованого підходу до інтелектуального аналізу логістичних даних, який поєднує методи кластеризації, класифікації та пошуку асоціативних правил із використанням OLAP-аналітики. Результатом роботи стала побудова повнофункціональної системи підтримки прийняття рішень, здатної забезпечувати комплексну оцінку, прогнозування та пояснення поведінки логістичних процесів транспортної компанії. У роботі створено сховище даних (Data Warehouse), що акумулює інформацію про клієнтів, транспортні засоби, маршрути та часові параметри виконання замовлень. На його основі розроблено багатовимірний OLAP-куб, який дав змогу оперативно виконувати аналітику показників доставки, розрахунок ключових KPI (On-Time Rate, Delay120 Ratio, Avg Delivery Time, Delivery Efficiency Score, Speed Index) та побудову інтерактивних звітів. Така архітектура забезпечила основу для реалізації аналітичних і прогнозних моделей Data Mining. Проведене дослідження продемонструвало, що застосування кластеризації дозволяє виокремити типові сценарії доставки з урахуванням особливостей маршрутів, умов перевезення та характеристик клієнтів. Використання класифікаційної моделі Наївного Байєса дало змогу передбачати ймовірність затримки доставки для кожної групи, а метод асоціативних правил дозволив пояснити причини таких результатів через виявлення закономірностей між факторами середовища. Такий комбінований підхід забезпечив не лише високу точність прогнозів, а й підвищену пояснюваність результатів, що є ключовим аспектом для практичного використання системи у процесі прийняття управлінських рішень. Результати експериментів показали, що для кластерів з більш однорідною структурою даних (наприклад, регулярні міські доставки) система демонструє вищу стабільність показників довіри та точності класифікації, тоді як для змішаних або складних сценаріїв (міжміські маршрути, вплив погодних умов) прогноз залишається контекстно залежним. Водночас поєднання класифікаційного аналізу з асоціативними правилами дозволило підтвердити понад 85% отриманих прогнозів, що свідчить про узгодженість результатів і достовірність аналітичних висновків. Ключовим досягненням роботи є підтвердження того, що інтеграція OLAP-аналітики та Data Mining створює новий рівень інтелектуальної підтримки управління логістикою. Система здатна не лише відображати стан показників, а й інтерпретувати причини їх змін, виявляючи приховані взаємозв'язки між факторами маршруту, рейтингу агентів, погодними умовами та типом транспортного засобу. Така функціональність забезпечує перехід від класичної звітності до моделі пояснюваної аналітики, що сприяє ухваленню більш обґрунтованих управлінських рішень. Практичні результати підтвердили ефективність системи у моніторингу логістичних KPI, прогнозуванні затримок та оптимізації маршрутів. Побудовані звіти продемонстрували стабільність

середнього часу доставки, зниження частки критичних затримок і зростання кількості своєчасних перевезень при збільшенні обсягу замовлень. Таким чином, розроблена система забезпечує керівництво транспортної компанії об'єктивною інформаційною основою для оцінки ефективності операцій і планування подальших дій.

7 СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. **Microsoft.** Online Analytical Processing (OLAP) — Overview. *Microsoft Learn*. URL: https://learn.microsoft.com/analysis-services/ (дата звернення: 12.11.2025).
2. **Han J., Kamber M., Pei J.** *Data Mining: Concepts and Techniques*. 3rd ed. Waltham : Morgan Kaufmann, 2012. 744 с.
3. **Kaggle.** Amazon Delivery Dataset. *Kaggle Datasets*. URL: https://www.kaggle.com/ (дата звернення: 12.11.2025).
4. **Microsoft.** STDistance (geography Data Type): Transact-SQL. *Microsoft Learn*. URL: https://learn.microsoft.com/ru-ru/sql/t-sql/spatial-geography/stdistance-geography-data-type?view=sql-server-ver17 (дата звернення: 12.11.2025).
5. **Kimball R., Ross M.** *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. 3rd ed. Hoboken : John Wiley & Sons, 2013. 608 с.
6. **Microsoft.** Key Performance Indicators (KPIs) — Multidimensional Models (SSAS). *Microsoft Learn*. URL: https://learn.microsoft.com/en-us/analysis-services/multidimensional-models/key-performance-indicators-kpis-in-multidimensional-models?view=sql-analysis-services-2025 (дата звернення: 12.11.2025).
7. **Microsoft.** Charts in paginated reports (SSRS). *Microsoft Learn*. URL: https://learn.microsoft.com/en-us/sql/reporting-services/report-design/charts-report-builder-and-ssrs?view=sql-server-ver17 (дата звернення: 12.11.2025).
8. **scikit-learn.** GaussianNB — API Reference. *Scikit-learn Documentation*. URL:https://scikitlearn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html (дата звернення: 12.11.2025).
9. **Manning C. D., Raghavan P., Schütze H.** *Introduction to Information Retrieval*. Cambridge : Cambridge University Press, 2008.
10. **scikit-learn.** Clustering — User Guide. *Scikit-learn Documentation*. URL: https://scikit-learn.org/stable/modules/clustering.html (дата звернення: 12.11.2025).
11. **Rousseeuw P. J.** Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*. 1987. Vol. 20. P. 53–65.
12. **Thorndike R. L.** Who Belongs in the Family? *Psychometrika*. 1953. Vol. 18, № 4. P. 267–276.
13. **scikit-learn.** KMeans — API Reference. *Scikit-learn Documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (дата звернення: 12.11.2025).

14. **Agrawal R., Srikant R.** Fast Algorithms for Mining Association Rules. *Proc. VLDB*. 1994. P. 487–499.

15. **mlxtend.** apriori — Frequent Itemsets. *mlxtend Documentation*. URL: http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/ (дата звернення: 12.11.2025).

16. **mlxtend.** association_rules — Rule Generation. *mlxtend Documentation*. URL:http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/ (дата звернення: 12.11.2025).

ДОДАТОК А

```
def main():
    # 1) Дані з VIEW
    with sql() as cn:
        df = pd.read_sql(f"SELECT * FROM {VIEW}", cn)

    # ключ
    key_col = "DeliveryKey" if "DeliveryKey" in df.columns else ("OrderID" if "OrderID" in df.columns else None)
    if key_col is None:
        raise KeyError("У VIEW немає DeliveryKey/OrderID — потрібен хоча б один ключ.")

    # інженерія ознак
    df["Pace_MinPerKm"] = np.where(
        pd.to_numeric(df["Distance_Km"], errors="coerce") > 0,
        pd.to_numeric(df["Delivery_Time"], errors="coerce") / pd.to_numeric(df["Distance_Km"], errors="coerce"),
        np.nan
    )

    num_cols = ["Distance_Km", "Delivery_Time", "Pace_MinPerKm", "AgentRating"]
    cat_cols = ["Area", "Vehicle", "Category", "Weather", "Traffic", "DayPart"]

    # ціль (4 класи)
    y = df["Delivery_Time"].apply(delay_band)
    mask = y != "unknown"
    df = df[mask].copy()
    y = y[mask].copy()

    # нормалізуємо ключ у числовий формат (для дойну з CSV)
    df[key_col] = pd.to_numeric(df[key_col], errors="coerce").astype("Int64")

    # 2) Пайплайн: імпутація → ohe/scaler → GaussianNB
    pre = ColumnTransformer(
        transformers=[
            ("num", Pipeline([("imp", SimpleImputer(strategy="median")),
                              ("sc", StandardScaler())]), num_cols),
            ("cat", Pipeline([("imp", SimpleImputer(strategy="most_frequent")),
                              ("ohe", make_ohe())]), cat_cols),
        ],
        remainder="drop"
    )
    clf = Pipeline(steps=[("pre", pre), ("nb", GaussianNB())])

    # 3) train/test split глобально (як у baseline)
    X = df[num_cols + cat_cols]
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42, stratify=y
    )

    # збережемо ключі для тесту (для дочойну кластерів)
    test_keys = df.loc[X_test.index, [key_col]].copy()

    # 4) тренування
    clf.fit(X_train, y_train)

    # 5) глобальні метрики на тесті (для контролю — мають бути близькі до baseline)
    y_pred = clf.predict(X_test)
    acc_global = accuracy_score(y_test, y_pred)
    macro_f1_global = f1_score(y_test, y_pred, average="macro")
    report_global = classification_report(y_test, y_pred, digits=3)
```

```
# 6) підтягнемо кластери (DeliveryKey, ClusterID_KMeans)
if not os.path.exists(KM_CSV):
    raise FileNotFoundError("Немає results/kmeans_clusters.csv — згенеруй файл KMeans із DeliveryKey, ClusterID_KMeans.")
km = pd.read_csv(KM_CSV)
if "DeliveryKey" not in km.columns and "OrderID" in km.columns:
    km = km.rename(columns={"OrderID": "DeliveryKey"})
if "DeliveryKey" not in km.columns:
    raise KeyError("kmeans_clusters.csv має містити DeliveryKey")
if "ClusterID_KMeans" not in km.columns and "ClusterID" in km.columns:
    km = km.rename(columns={"ClusterID": "ClusterID_KMeans"})
if "ClusterID_KMeans" not in km.columns:
    raise KeyError("kmeans_clusters.csv має містити ClusterID_KMeans")

km["DeliveryKey"] = pd.to_numeric(km["DeliveryKey"], errors="coerce").astype("Int64")

# 7) зберемо тестову таблицю з кластерами та прогнозами
test_df = pd.DataFrame({
    key_col: test_keys[key_col].values,
    "y_true": y_test.values,
    "y_pred": y_pred
})
# нормалізуємо ім'я ключа, щоби змерджити з km (який має DeliveryKey)
if key_col != "DeliveryKey":
    test_df = test_df.rename(columns={key_col: "DeliveryKey"})
test_df["DeliveryKey"] = pd.to_numeric(test_df["DeliveryKey"], errors="coerce").astype("Int64")

merged = test_df.merge(km[["DeliveryKey", "ClusterID_KMeans"]], on="DeliveryKey", how="inner")

# 8) пер-кластерні метрики
rows = []
for k, g in merged.groupby("ClusterID_KMeans"):
    acc = accuracy_score(g["y_true"], g["y_pred"])
    m_f1 = f1_score(g["y_true"], g["y_pred"], average="macro")
    rows.append({"ClusterID_KMeans": int(k), "size": int(len(g)), "accuracy": acc, "macro_f1": m_f1})

scores = pd.DataFrame(rows).sort_values("ClusterID_KMeans").reset_index(drop=True)
# агрегати
valid = scores.dropna(subset=["macro_f1"])
mean_macro = float(valid["macro_f1"].mean()) if not valid.empty else np.nan
med_macro  = float(valid["macro_f1"].median()) if not valid.empty else np.nan
w_macro    = float(np.average(valid["macro_f1"], weights=valid["size"])) if not valid.empty else np.nan

summary = {
    "global_test_accuracy": float(acc_global),
    "global_test_macro_f1": float(macro_f1_global),
    "clusters_evaluated": int(valid.shape[0]),
    "mean_macro_f1": mean_macro,
    "median_macro_f1": med_macro,
    "weighted_macro_f1": w_macro
}
```

ДОДАТОК Б

```python
# -------------------- майн правил --------------------
def mine_assoc_rules(transactions: list, min_supp: float, min_conf: float, min_lift: float) -> pd.DataFrame:
    try:
        from mlxtend.frequent_patterns import apriori, association_rules
    except Exception as e:
        raise RuntimeError("Потрібна бібліотека mlxtend (pip install mlxtend). Деталі: " + str(e))

    all_items = sorted(set().union(*transactions)) if transactions else []
    if not all_items:
        return pd.DataFrame(columns=["Antecedents","Consequents","Support","Confidence","Lift"])

    data = []
    for t in transactions:
        row = {item: (1 if item in t else 0) for item in all_items}
        data.append(row)
    basket = pd.DataFrame(data, dtype=int)

    freq = apriori(basket, min_support=min_supp, use_colnames=True)
    if freq.empty:
        return pd.DataFrame(columns=["Antecedents","Consequents","Support","Confidence","Lift"])

    rules = association_rules(freq, metric="confidence", min_threshold=min_conf)
    if rules.empty:
        return pd.DataFrame(columns=["Antecedents","Consequents","Support","Confidence","Lift"])

    rules = rules[rules["lift"] >= min_lift].copy()
    if rules.empty:
        return pd.DataFrame(columns=["Antecedents","Consequents","Support","Confidence","Lift"])

    rules = rules[rules["consequents"].apply(lambda s: any(str(x).startswith("delayband=") for x in s))].copy()
    if rules.empty:
        return pd.DataFrame(columns=["Antecedents","Consequents","Support","Confidence","Lift"])

    def frozenset_to_csv(fs):
        return ",".join(sorted(clean_tok(x) for x in fs))

    out = pd.DataFrame({
        "Antecedents": rules["antecedents"].apply(frozenset_to_csv),
        "Consequents": rules["consequents"].apply(frozenset_to_csv),
        "Support":     rules["support"].astype(float),
        "Confidence":  rules["confidence"].astype(float),
        "Lift":        rules["lift"].astype(float)
    })
    out = out[(out["Antecedents"].str.len() > 0) & (out["Consequents"].str.len() > 0)]
    out = out.drop_duplicates().reset_index(drop=True)
    return out
```

```python
def main():
    # 1) Дані
    with sql() as cn:
        df = pd.read_sql(f"SELECT * FROM {VIEW}", cn)

    df["Pace_MinPerKm"] = np.where(
        pd.to_numeric(df["Distance_Km"], errors="coerce") > 0,
        pd.to_numeric(df["Delivery_Time"], errors="coerce") / pd.to_numeric(df["Distance_Km"], errors="
        np.nan
    )

    num_cols = ["Distance_Km", "Delivery_Time", "Pace_MinPerKm", "AgentRating"]
    cat_cols = ["Area", "Vehicle", "Category", "Weather", "Traffic", "DayPart"]

    y = df["Delivery_Time"].apply(delay_band)
    mask = y != "unknown"
    df, y = df[mask].copy(), y[mask].copy()

    key_col = "DeliveryKey" if "DeliveryKey" in df.columns else ("OrderID" if "OrderID" in df.columns e
    if key_col is None:
        raise KeyError("VIEW must contain DeliveryKey or OrderID")
    df[key_col] = pd.to_numeric(df[key_col], errors="coerce").astype("Int64")

    pre = ColumnTransformer(
        transformers=[
            ("num", Pipeline([("imp", SimpleImputer(strategy="median")),
                              ("sc",  StandardScaler())]), num_cols),
            ("cat", Pipeline([("imp", SimpleImputer(strategy="most_frequent")),
                              ("ohe", make_ohe())]), cat_cols),
        ], remainder="drop"
    )
    clf = Pipeline(steps=[("pre", pre), ("nb", GaussianNB())])

    X = df[num_cols + cat_cols]
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42, stratify=y
    )

    cols_for_explain = [key_col] + cat_cols + ["AgentRating"]
    test_frame = df.loc[X_test.index, cols_for_explain].copy()

    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)

    # ймовірність саме передбаченого класу
    proba = clf.predict_proba(X_test)
    classes = clf.named_steps["nb"].classes_
    # для кожного рядка беремо proba відповідного y_pred
    pred_probs = []
    for i, cls in enumerate(y_pred):
        j = int(np.where(classes == cls)[0][0])
        pred_probs.append(float(proba[i, j]))

    acc = accuracy_score(y_test, y_pred)
    macro_f1 = f1_score(y_test, y_pred, average="macro")
    print(f"[NB] test Acc={acc:.3f} | MacroF1={macro_f1:.3f}")
    print(classification_report(y_test, y_pred, digits=3))

    pred_df = test_frame.copy()
    pred_df["PredDelayBand_NB"] = y_pred
    pred_df["PredProb"] = pred_probs
    pred_csv = os.path.join(OUT_DIR, "nb_test_predictions.csv")
    pred_df.to_csv(pred_csv, index=False, encoding="utf-8-sig")
    print("Saved:", pred_csv)
```

```python
# 3) Транзакції для правил
transactions = []
delay_labels = y.loc[X.index]
full_for_tokens = df.loc[X.index, cols_for_explain + ["Delivery_Time"]].copy()
full_delay = delay_labels.apply(lambda z: f"delayband={z}")

for i, row in full_for_tokens.iterrows():
    toks = tokens_for_row_base(row)
    toks = set(toks)
    toks.add(clean_tok(full_delay.loc[i]))
    transactions.append(toks)

# 4) Майнимо правила
rules_df = mine_assoc_rules(transactions, MIN_SUPP, MIN_CONF, MIN_LIFT)
rules_path = os.path.join(OUT_DIR, "assoc_rules.csv")
rules_df.to_csv(rules_path, index=False, encoding="utf-8-sig")
print(f"Saved rules: {rules_path} ({len(rules_df)} rows)")

# 5) Індекси правил по наслідку
buckets = {}
for i, r in rules_df.iterrows():
    for cons in r["Consequents"].split(","):
        cl = clean_tok(cons)
        if cl.startswith("delayband="):
            buckets.setdefault(cl, []).append(i)

# 6) Пояснення NB-прогнозів
rows, matched = [], 0
for idx, (_, rec) in enumerate(pred_df.iterrows()):
    pred = str(rec["PredDelayBand_NB"]).strip()
    cons_key = f"delayband={pred}".lower()
    cand_idx = set(buckets.get(cons_key, []))
    toks = tokens_for_row_base(rec)

    best_idx, best_score = None, None
    for ridx in cand_idx:
        ante = set(clean_tok(x) for x in rules_df.loc[ridx, "Antecedents"].split(",") if x.strip())
        if ante.issubset(toks):
            conf = safe_num(rules_df.loc[ridx, "Confidence"])
            lift = safe_num(rules_df.loc[ridx, "Lift"])
            score = (conf, lift, len(ante))
            if best_score is None or score > best_score:
                best_score, best_idx = score, ridx
    if best_idx is not None:
        matched += 1
        rows.append({
            key_col: int(rec[key_col]) if pd.notna(rec[key_col]) else None,
            "PredDelayBand_NB": pred,
            "PredProb": float(rec["PredProb"]),
            "TopRule_Ante": rules_df.loc[best_idx, "Antecedents"],
            "TopRule_Cons": rules_df.loc[best_idx, "Consequents"],
            "TopRule_Support": float(rules_df.loc[best_idx, "Support"]),
            "TopRule_Conf": float(rules_df.loc[best_idx, "Confidence"]),
            "TopRule_Lift": float(rules_df.loc[best_idx, "Lift"]),
            "MatchLen": int(len(rules_df.loc[best_idx, "Antecedents"].split(",")))
        })
    else:
        rows.append({
            key_col: int(rec[key_col]) if pd.notna(rec[key_col]) else None,
            "PredDelayBand_NB": pred,
            "PredProb": float(rec["PredProb"]),
            "TopRule_Ante": "",
            "TopRule_Cons": "",
            "TopRule_Support": np.nan,
            "TopRule_Conf": np.nan,
            "TopRule_Lift": np.nan,
            "MatchLen": 0
        })

explain_df = pd.DataFrame(rows)
out_path = os.path.join(OUT_DIR, "nb_assoc_explanations.csv")
explain_df.to_csv(out_path, index=False, encoding="utf-8-sig")
print("Saved:", out_path)

coverage = 100.0 * matched / len(pred_df) if len(pred_df) else 0.0
summary = {
    "nb_test_rows": int(len(pred_df)),
    "explained_rows": int(matched),
    "coverage_pct": round(coverage, 2),
    "nb_test_acc": round(float(acc), 3),
    "nb_test_macro_f1": round(float(macro_f1), 3),
    "assoc_rules": {
        "min_support": MIN_SUPP,
        "min_confidence": MIN_CONF,
        "min_lift": MIN_LIFT,
        "total_rules": int(len(rules_df))
    },
    "note": "Правила будуються на тих самих ознаках, наслідок лише delayband=*. Пояснення NB шукає правило з цим наслідком, антецедент ⊆ токенів рядка."
}
with open(os.path.join(OUT_DIR, "nb_assoc_explain_summary.json"), "w", encoding="utf-8") as f:
    json.dump(summary, f, ensure_ascii=False, indent=2)
print("Summary:", summary)
```

ДОДАТОК В

```python
# -------- ОСНОВНА ЛОГІКА --------
def main():
    # 1) Зчитуємо kmeans_clusters.csv (ключ = OrderID )
    if not os.path.exists(KMEANS_CSV):
        raise FileNotFoundError(f"Не знайдено файл кластерів: {KMEANS_CSV}")

    km = pd.read_csv(KMEANS_CSV)
    # Визначаємо назву колонки ключа та кластеру
    key_col_km = None
    for cand in ["OrderID",]:
        if cand in km.columns:
            key_col_km = cand
            break
    if key_col_km is None:
        raise ValueError("У kmeans_clusters.csv відсутній ключ (OrderID / DeliveryKey).")

    cluster_col = None
    for cand in ["ClusterID_KMeans", "cluster", "cluster_id", "ClusterID"]:
        if cand in km.columns:
            cluster_col = cand
            break
    if cluster_col is None:
        raise ValueError("У kmeans_clusters.csv відсутня колонка з кластером (ClusterID_KMeans).")

    km_key = to_num_int(km[key_col_km])
    km = km.assign(_key=km_key)[["_key", cluster_col]].dropna()
    km.rename(columns={cluster_col: "ClusterID_KMeans"}, inplace=True)

    # 2) Тягнемо дані з VIEW (ключ = DeliveryKey)
    with sql() as cn:
        df = pd.read_sql(f"SELECT * FROM {VIEW}", cn)

    if "DeliveryKey" not in df.columns:
        raise KeyError(f"{VIEW} має містити колонку DeliveryKey")

    df["DeliveryKey"] = to_num_int(df["DeliveryKey"])

    # Базові колонки для правил
    needed = ["DeliveryKey", "Area", "Vehicle", "Category", "Weather", "Traffic", "DayPart",
              "AgentRating", "Delivery_Time"]
    miss = [c for c in needed if c not in df.columns]
    if miss:
        raise KeyError(f"У {VIEW} відсутні обов'язкові колонки: {miss}")

    df = df[needed].copy()

    # 3) Мердж DeliveryKey (БД) ↔ OrderID/DeliveryKey (CSV)
    merged = df.merge(km, left_on="DeliveryKey", right_on="_key", how="inner")
    if merged.empty:
        raise ValueError("Після мерджу з кластерами отримали 0 рядків. "
                         "Перевір, що DeliveryKey (VIEW) збігається з OrderID у kmeans_clusters.csv (як числа).")
```

```python
    # 4) Групуємо за кластерами, формуємо транзакції та видобуваємо правила
    all_rules = []
    summary_rows = []

    for cluster_id, g in merged.groupby("ClusterID_KMeans"):
        # формуємо транзакції (список списків токенів)
        transactions = g.apply(build_tokens, axis=1).tolist()
        n = len(transactions)
        min_supp = dynamic_support(n)

        rules = mine_rules_for_cluster(transactions, min_supp=min_supp)
        rules["ClusterID_KMeans"] = cluster_id
        rules["cluster_size"] = n
        rules["min_support_used"] = min_supp

        all_rules.append(rules)

        # статистика по кластеру
        top = rules.head(TOP_N_PER_CLUSTER).copy() if not rules.empty else pd.DataFrame()
        top_json = top[["antecedents", "consequents", "support", "confidence", "lift"]].to_dict(orient="records")

        summary_rows.append({
            "ClusterID_KMeans": int(cluster_id),
            "cluster_size": int(n),
            "min_support_used": float(min_supp),
            "rules_found": int(len(rules)),
            "top_rules": top_json
        })

    # 5) Збереження
    if all_rules:
        rules_full = pd.concat(all_rules, ignore_index=True)
    else:
        rules_full = pd.DataFrame(columns=[
            "antecedents", "consequents", "support", "confidence", "lift",
            "ClusterID_KMeans", "cluster_size", "min_support_used"
        ])

    full_csv = os.path.join(OUT_DIR, "cluster_rules_full.csv")
    rules_full.to_csv(full_csv, index=False, encoding="utf-8-sig")
    print(f"Saved: {full_csv} ({len(rules_full)} rows)")

    summary_df = pd.DataFrame(summary_rows).sort_values("ClusterID_KMeans")
    summ_csv = os.path.join(OUT_DIR, "cluster_rules_summary.csv")
    summary_df.to_csv(summ_csv, index=False, encoding="utf-8-sig")
    print(f"Saved: {summ_csv}")

    # JSON зручний для швидкого перегляду у звіті / UI
    summ_json = os.path.join(OUT_DIR, "cluster_rules.json")
    with open(summ_json, "w", encoding="utf-8") as f:
        json.dump(summary_rows, f, ensure_ascii=False, indent=2)
    print(f"Saved: {summ_json}")
```