МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ІНТЕЛЕКТУАЛЬНА АНАЛІТИЧНА СИСТЕМА ДЛЯ ПРОГНОЗУВАННЯ ПОПИТУ НА ТОВАРИ В E-COMMERCE»

Виконав: студент 2 курсу

рівень: магістр

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп’ютерні науки»

Дроздов Богдан Михайлович

Керівник: к.пед.н., доцент

Лобода Юлія Геннадіївна

Рецензент: к.т.н., доцент

Трофименко Олена Григорівна

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет **кібербезпеки та інформаційних технологій**
Кафедра **інформаційних технологій**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **122 «Комп'ютерні науки»**
Назва освітньої програми: **Комп'ютерні науки**

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних технологій
доцент _____ Н.І. Логінова
«____ »_____ 20__ року

ЗАВДАННЯ
на кваліфікаційну (магістерську) роботу
здобувачу Дроздову Богдану Михайловичу

1. Тема роботи: «Інтелектуальна аналітична система для прогнозування попиту на товари в e-commerce»

Керівник роботи: к.пед.н, доцент Лобода Юлія Геннадіївна
затверджені наказом НУ ОЮА від «*10*» *жовтня 2025* року № *3182-06*

2. Строк подання здобувачем роботи «*25*» *листопада 2025 рік*

3. Дата видачі завдання «*02*» *червня 2025 рік*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми, вивчення літературних джерел та складання плану роботи	09.09.2025	
2.	Підготовка першого розділу роботи та подання його керівнику	01.10.2025	
3.	Підготовка другого розділу роботи та подання його керівнику	14.10.2025	
4.	Підготовка третього розділу роботи та подання його керівнику	07.11.2025	
5.	Підготовка вступу та висновків	10.11.2025	
6.	Доопрацювання роботи з урахуванням зауважень керівника	24.11.2025	
7.	Подача електронного варіанту роботи для перевірки на плагіат	25.11.2025	
8.	Допуск роботи до захисту завідуючим кафедри	01.12.2025	
9.	Захист роботи	03.12.2025	

Здобувач _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інтелектуальна аналітична система для прогнозування попиту на товари в e-commerce» на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 122 Комп'ютерні науки, освітньою програмою: Комп'ютерні науки, містить 16 рисунків, 29 літературних джерела за переліком посилань. Робота виконана на 63 сторінках загального тексту і 60 сторінках основного тексту.

Метою дослідження є розробка концепції інтелектуальної аналітичної системи для прогнозування попиту на товари електронної комерції, яка забезпечує підвищення точності прогнозів шляхом застосування сучасних методів аналізу даних, машинного навчання та автоматизації процесів обробки інформації .

Об'єктом дослідження є процеси прогнозування попиту в системах електронної комерції.

Предметом дослідження – методи, моделі та аналітичні підходи машинного навчання, що використовуються для прогнозування попиту та побудови інтелектуальних аналітичних систем у сфері електронної комерції.

У кваліфікаційній роботі розглядаються особливості формування попиту на товари електронна комерції, сучасні підходи до прогнозування та класифікація аналітичних систем. Проаналізовано методи машинного навчання, статистичні моделі та інструменти багатовимірного моделювання попиту. Розроблено архітектурну концепцію інтелектуальної аналітичної системи, яка включає компоненти збору, обробки та аналітичного опрацювання даних. Визначено фактори, що найбільше впливають на точність прогнозу. Сформульовано рекомендації щодо впровадження систем прогнозування попиту у бізнес-процеси компаній електронної комерції.

ПРОГНОЗУВАННЯ ПОПИТУ, ЕЛЕКТРОННА КОМЕРЦІЯ, МАШИННЕ НАВЧАННЯ, ІНТЕЛЕКТУАЛЬНІ АНАЛІТИЧНІ СИСТЕМИ, МОДЕЛІ ПРОГНОЗУВАННЯ, АНАЛІЗ ДАНИХ

ABSTRACT

The qualification thesis titled “Intelligent analytical system for forecasting demand for goods in e-commerce”, submitted for the second (master’s) level of higher education in the specialty 122 Computer Science, educational program: Computer Science, contains 16 figures and 29 references listed in the bibliography. The thesis consists of 63 pages of total text and 60 pages of main content.

The purpose of the research is to develop a concept of an intelligent analytical system for forecasting demand for goods in electronic commerce, which ensures improved prediction accuracy through the use of modern data analysis methods, machine learning, and automation of data processing procedures.

The object of the research is the processes of demand forecasting in electronic commerce systems.

The subject of the research is the methods, models, and analytical machine learning approaches used for demand forecasting and the development of intelligent analytical systems in the field of electronic commerce.

The qualification thesis examines the peculiarities of demand formation in electronic commerce, modern forecasting approaches, and the classification of analytical systems. Machine learning methods, statistical models, and tools for multidimensional demand modelling are analyzed. An architectural concept of an intelligent analytical system is developed, which includes components for data collection, processing, and analytical computation. The factors that have the greatest influence on forecasting accuracy are identified. Recommendations for implementing demand forecasting systems into the business processes of e-commerce companies are formulated.

DEMAND FORECASTING, E-COMMERCE, MACHINE LEARNING, INTELLIGENT ANALYTICAL SYSTEMS, FORECASTING MODELS, DATA ANALYSIS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

ARIMA – AutoRegressive Integrated Moving Average

ACID - Atomicity, Consistency, Isolation, Durability

LSTM – Long Short-Term Memory

MAE – Mean Absolute Error

MAPE – Mean Absolute Percentage Error

ML – Machine Learning

RMSE – Root Mean Square Error

ETL – Extract, Transform, Load

SQL – Structured Query Language

NoSQL – Not Only SQL

ERP - Enterprise Resource Planning

WMS - Warehouse Management System

OMS - Order Management System

AI - Artificial Intelligence

MLOps - Machine Learning Operations

ETS - Exponential Smoothing State Space Model

JSON - JavaScript Object Notation

TTL - Time To Live

OLAP - Online Analytical Processing

OLTP - Online Transaction Processing

БД – бази даних

СКБД – системи керування базами даних

ВСТУП

Актуальність теми дослідження зумовлена стрімким розвитком електронної комерції, де точність прогнозування попиту стає ключовим фактором ефективності операційної діяльності. Умови високої конкуренції, динамічні зміни асортименту, сезонність та вплив зовнішніх подій формують складне середовище, у якому традиційні методи прогнозування демонструють недостатню гнучкість. Сучасні підходи машинного навчання, автоматизація процесів аналізу даних та створення інтелектуальних аналітичних систем дозволяють значно підвищити точність передбачень і приймати обґрунтовані управлінські рішення в електронної комерції.

Метою дослідження – розробка концепції інтелектуальної аналітичної системи для прогнозування попиту на товари електронної комерції, яка забезпечує підвищення точності прогнозів шляхом застосування сучасних методів аналізу даних, машинного навчання та автоматизації процесів обробки інформації.

Об'єктом дослідження є процеси прогнозування попиту в системах електронної комерції.

Предметом дослідження є методи, моделі та аналітичні підходи машинного навчання, що використовуються для прогнозування попиту та побудови інтелектуальних аналітичних систем у сфері електронної комерції.

Для досягнення поставленої мети в кваліфікаційній роботі необхідно вирішити наступні **завдання**:

– проаналізувати предметну область електронної комерції та специфіку формування попиту на товари;

– дослідити існуючі методи та моделі прогнозування попиту та визначити оптимальні підходи для e-commerce;

– розробити архітектурну концепцію інтелектуальної аналітичної системи прогнозування попит;

– розробити та навчити моделі машинного навчання для прогнозування попиту;

– провести порівняльний аналіз точності моделей за ключовими метриками;

– сформулювати рекомендації щодо впровадження системи прогнозування попиту в реальні бізнес-процеси електронної комерції.

В кваліфікаційній роботі використовувалися загальнонаукові та спеціальні методи наукового дослідження таки, як системний аналіз, синтез, порівняння, статистичний аналіз, методи машинного навчання, класифікація, візуалізація даних тощо.

Теоретична значущість полягає в узагальненні та систематизації методів прогнозування попиту, розкритті ролі інженерії ознак у задачах прогнозування та обґрунтуванні архітектури інтелектуальних аналітичних систем.

Практичне значення полягає в можливості використання отриманих результатів для побудови прикладних аналітичних рішень у сфері електронної комерції, підвищення точності прогнозування попиту, оптимізації запасів та підтримки управлінських рішень у компаніях роздрібної торгівлі.

Результати кваліфікаційного дослідження апробовані на VI Міжнародної науково-практичної конференції «Кібербезпека в сучасному світі: актуальні виклики», (м. Одеса, 28 листопада 2025 р.).

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ПОПИТУ В ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1 Особливості формування попиту на товари електронної комерції

У цифрових каналах купівлі-продажу попит формується в умовах, де рішення клієнта підсилюється або змінюється рекомендаційними механізмами, рекламою, відгуками, швидкими промокампаніями та цінами, що можуть змінюватися майже в реальному часі. Як наслідок, попит в електронної комерції часто проявляє різкі сплески та «злами» трендів, а також суттєву неоднорідність між товарами та категоріями [1]. У систематичному огляді моделей прогнозування попиту для ритейлу електронної комерції підкреслюється, що онлайн-ринок характеризується волатильною поведінкою споживачів і різноманітним асортиментом, а прогнозування стає стратегічною функцією для управління запасами й доставкою.

На відміну від офлайн-ритейлу, де сезонність часто задає «каркас» попиту, в електронної комерції попит суттєво чутливий до короткострокових стимулів: флеш-розпродажів, кампаній інфлюенсерів і стратегій динамічного ціноутворення. Саме ці фактори формують нелінійні, складно передбачувані сплески [2].

Додатково впливають «миттєві» інформаційні хвилі: віральність у цифрових каналах підсилює ефект зовнішніх шоків (наприклад, пандемій), що може давати більш різку реакцію попиту онлайн порівняно з традиційними каналами.

Попит в електронної комерції формується не лише через фактичні продажі, а й через передумови купівлі, які фіксуються цифровими слідами [3]. У рамках огляду акцентується, що поведінкові потоки (перегляди, додавання в кошик, тривалість сесії) є оперативними сигналами наміру покупки та зародження трендів.

Така природа попиту означає, що модель має бачити не тільки історію продажів, а й «воронку» поведінки, де попит зароджується значно раніше, ніж відбувається транзакція.

У великих маркетплейсах та інтернет-магазинах прогнозування ускладнюється через велику кількість товарних позицій, швидке оновлення асортименту та роботу кількох центрів оброблення й відвантаження замовлень; окремо підкреслюється, що розширення номенклатури та гнучке (динамічне) ціноутворення підвищують складність логістики й потребують більш адаптивних підходів до прогнозування [4,5].

Фактично попит формується як сума багатьох «локальних» процесів на рівні товарного асортименту, де кожен товар має власний життєвий цикл, чутливість до акцій та цінових змін.

На практиці фактори попиту в електронної комерції доцільно агрегувати в кілька груп:

- календарно-часові: день тижня, свята, сезонність, лаги продажів. У огляді підкреслено важливість темпоральних ознак (день тижня, сезонність, свята) для відтворення циклічних компонент;
- промо та маркетинг: знижки, кампанії, флеш-розпродажі, маркетингові витрати. Для електронної комерції прямо зазначено вирішальність функцій, пов'язаних з просуванням, які описують коливання попиту;
- поведінкові: перегляди товару, додавання в кошик, сесійні метрики та інші «індикатори наміру»;
- цінові: абсолютна ціна, відносна ціна (відносно категорії/аналогів), ознаки динамічного ціноутворення;
- зовнішні: ринкові тренди, макроподії, попитні індекси/тренди пошуку.

Через різнорідність сигналів найбільший вплив на якість прогнозування часто має не вибір алгоритму, а коректне збирання та перетворення даних. У дослідженнях, присвячених прогнозуванню трендів в електронній комерції [6], , окремо виокремлюють етапи попередньої обробки (очищення даних, робота з пропусками та викидами), нормалізацію (зокрема масштабування до заданого діапазону) та подальшу побудову ознак для врахування сезонних коливань і впливу промоактивностей[7].

1.2 Підходи до прогнозування попиту в електронної комерції та їх класифікація

Ураховуючи специфіку формування попиту в електронній комерції (високу мінливість, промо-сплески, швидке оновлення асортименту та наявність поведінкових сигналів), підходи до прогнозування доцільно класифікувати за кількома взаємодоповнювальними ознаками [8]. Така класифікація дає змогу обґрунтовано вибирати метод моделювання відповідно до бізнес-цілей (управління запасами, логістика, маркетинг), доступних даних і масштабу системи.

1) За типом математичного апарату (модельним класом):

– статистичні часові моделі (класичні методи часових рядів) використовуються як базові та добре інтерпретуються, однак можуть втрачати точність за різких зламів тренду та промо-аномалій.

– методи машинного навчання краще відпрацьовують нелінійні залежності та взаємодію чинників (ціна, промо, поведінкові метрики), а також придатні для масштабування на великий асортимент.

– методи глибинного навчання орієнтовані на складні часові залежності та багатовимірні потоки даних; доцільні за наявності достатніх обсягів якісної історії та стабільного контуру збирання ознак.

– гібридні та ансамблеві підходи поєднують сильні сторони різних моделей (наприклад, виокремлення тренду/сезонності + ML для нелінійних ефектів), що часто підвищує стійкість до промо-сплесків і шуму.

2) За рівнем деталізації прогнозу (рівнем агрегації):

– на рівні товарної позиції прогноз є найбільш корисним для оперативного управління запасами, однак ускладняються розрідженістю продажів і ефектом «довгого хвоста» в асортименті.

– на рівні категорії/бренду/складу/регіону часові ряди більш стабільні та зручні для планування, але менш придатні для точного поповнення конкретних позицій.

– ієрархічне прогнозування забезпечує узгодження прогнозів між рівнями (позиція → категорія → загальний попит), що важливо для системного управління портфелем товарів.

3) За використанням додаткових факторів (джерел сигналів):

– уніваріантні підходи (лише історія продажів) простіші, але часто недостатні в умовах e-commerce.

– мультиваріантні підходи (із зовнішніми/екзогенними змінними) враховують ціни, промоактивності, календарні ефекти, маркетингові кампанії та поведінкові метрики, що робить прогноз чутливим до реальних драйверів попиту.

4) За горизонтом прогнозування:

– короткострокові (дні/тижні) потрібні для поповнення запасів і планування доставки та є найбільш чутливими до промо й операційних змін.

– середньострокові (тижні/місяці) використовуються для планування закупівель і маркетингових активностей.

– довгострокові (квартали) підтримують стратегічне планування асортименту та бюджетування, але мають вищу невизначеність.

5) За умовами наявності історії (типом задачі):

– звичайне прогнозування застосовується для товарів із достатньою історією продажів.

– прогнозування для нових товарів (cold start) спирається на характеристики товару, категорійні ознаки, аналоги, ціновий рівень, промо-плани та ранні поведінкові сигнали (перегляди, додавання в кошик).

6) За способом формування прогнозу (стратегія побудови):

– прямий прогноз (direct): будується окрема модель на кожен горизонт (t+1, t+7, t+30); підхід стійкіший до різних режимів попиту, але складніший у супроводі.

– рекурсивний (recursive): модель прогнозує один крок уперед, а далі прогноз використовується як вхід для наступних кроків; простіший, але може накопичувати помилку на довгих горизонтах.

– багатовихідний (multi-horizon): одна модель одразу формує вектор прогнозів на кілька кроків, що зручно для планування та узгодження рішень.

7) За масштабом навчання (організація моделей для багатьох товарів):

– окремі моделі для кожної позиції можуть забезпечувати високу точність для ключових товарів, але погано масштабуються на великий каталог.

– глобальна модель для групи/категорії навчається на даних багатьох товарів, переносить знання між схожими позиціями та краще працює з «довгим хвостом».

– сегментні (кластеризовані) моделі передбачають попередню сегментацію товарів за характером попиту та подальше моделювання в межах сегментів, що дозволяє збалансувати точність і трудомісткість.

8) За характером попиту (типом часового ряду) доцільно виділяти попит стабільний, трендовий, сезонний, промо-чутливий та переривчастий (із частими нульовими продажами). Для кожного типу доцільні різні класи моделей і різні правила побудови ознак.

Таблиця 1.1 – Рекомендовані підходи прогнозування залежно від типу попиту

Тип даних / попиту	Рекомендовані підходи	Пояснення
Регулярний попит	Holt–Winters, SARIMA	Тренд і сезонність виражені, модель добре інтерпретується
Сильна сезонність (кілька циклів)	Prophet, STL-підхід	Зручно описувати сезонні компоненти та календарні ефекти
Попит, чутливий до промо та ціни	Моделі ML (зокрема градієнтне підсилення)	Потрібно враховувати багато факторів і їх взаємодію
Нелінійні патерни й довгі залежності	LSTM, GRU, Transformer-підходи	Краще відтворюють складні часові взаємозв’язки
Переривчастий попит (часті нулі)	Croston, TSB та подібні	Спеціалізовані методи для розріджених продажів

Отже, прогнозування попиту в електронній комерції доцільно розглядати як комплексну задачу, що поєднує часовий аналіз і моделювання впливу зовнішніх чинників. Тому жоден окремий метод не є універсальним: базові моделі забезпечують еталон для порівняння, статистичні методи – інтерпретованість, ML-моделі – гнучкість у роботі з багатьма факторами, а глибоке навчання – здатність відтворювати складні

залежності. На практиці найбільш ефективними виявляються комбіновані підходи, де вибір моделі визначається характером попиту, наявністю факторів, горизонтом прогнозування та рівнем агрегації даних.

1.3 Аналітичні системи прогнозування та сучасні підходи до автоматизації

Удосконалення прогнозування попиту є ключовим напрямом розвитку електронної комерції, оскільки точність прогнозів безпосередньо впливає на управління запасами, логістичну ефективність і рівень обслуговування клієнтів. Сучасні аналітичні системи прогнозування попиту еволюціонували від локальних розрахунків на основі часових моделей до інтегрованих цифрових екосистем, що поєднують керування даними, машинне навчання, автоматизацію життєвого циклу моделей та взаємодію з ERP/WMS/OMS. Практика промислового впровадження підкреслює, що підтримка якості прогнозів залежить не лише від алгоритмів, а й від архітектури системи та керованості ML-компонентів [9].

З огляду на часові горизонти планування і управлінські задачі доцільно виокремлювати три класи систем.

1. Операційні системи прогнозування (оперативний рівень).

Орієнтовані на щоденні рішення: поповнення запасів, короткострокове планування, управління розміщенням запасів у мережі складів. Зазвичай працюють з високою частотою оновлення прогнозів (денні/внутрішньоденні цикли) та інтеграцією з операційними контурами (ERP/WMS/OMS). Високою є роль оперативних індикаторів і даних про запаси: зокрема, результати досліджень показують важливість поєднання короткострокових прогнозів із даними про поточні запаси для зменшення втрат продажів через дефіцит [10].

2. Тактичні системи планування (середньостроковий рівень).

Використовуються для планування промокампаній, сезонних стратегій продажів, закупівель та бюджетування на горизонтах від кількох тижнів до кількох місяців. На

цьому рівні ключовими стають модулі аналізу сезонності, промоефектів, цінової чутливості, а також оцінювання сценаріїв (наприклад, «що буде, якщо змінити глибину знижки/частоту акцій»).

3. Стратегічні аналітичні платформи (довгостроковий рівень).

Застосовуються для сценарного моделювання, довгострокової оцінки попиту, планування потужностей (capacity planning) та узгодження інвестиційних рішень у ланцюгах постачання. Такі платформи часто включають оптимізаційні модулі та засоби оцінювання узгодженості прогнозів на різних рівнях агрегування (ієрархічне прогнозування і узгодження).

Більшість промислових рішень спираються на такі архітектурні підходи.

1) Хмарні аналітичні платформи як базова інфраструктура. Хмарна модель спрощує масштабування обчислень і зберігання, що є критичним для електронної комерції з великим асортиментом і нерівномірним навантаженням. Загальна тенденція зростання витрат на публічну хмару та поширення гібридних підходів підтверджується прогнозами Gartner [11].

Типовий стек включає: сховище даних (Data Lake / Data Warehouse), інструменти розподіленої обробки та платформи для навчання/розгортання моделей. Приклади сервісів: Amazon S3/Redshift/SageMaker, Google Cloud Storage/BigQuery/Vertex AI, Azure Blob/Synapse/Azure ML.

2) MLOps як механізм автоматизації життєвого циклу моделей. Для забезпечення відтворюваності та керованості моделей прогнозування потрібні конвеєри підготовки ознак, реєстрація моделей, контроль якості та моніторинг деградації. Концептуально це узгоджується з підходом до зменшення «технічного боргу» в ML-системах: без дисципліни експлуатації витрати на підтримку якості прогнозів швидко зростають [9].

Типові модулі MLOps у системах прогнозування попиту:

– перевірка якості даних та валідація наборів;
– конвеєри побудови ознак;
– навчання/перенавчання (за розкладом або за сигналами з моніторингу);

– реєстр моделей і керування версіями;

– моніторинг якості прогнозів і зсуву даних (drift). Приклад, MLflow Model Registry [12] .

3) Архітектури обробки “майже в реальному часі”. Для високодинамічних категорій (швидка мода, продукти, маркетплейси з частими акціями) важливим є оперативне оновлення ознак і прогнозів у коротких циклах. На практиці це реалізується через потокове надходження подій (перегляди/кошик/ціни/залишки), експлуатацію прогнозних сервісів через API та часте перерахування прогнозів. Промислові кейси масштабування прогнозування й експлуатації моделей демонструють, зокрема, матеріали Zalando про організацію прогнозних конвеєрів та MLOps-процеси [13].

Корпоративні практики: узагальнений аналіз підходів

1. Ансамблювання та комбінування прогнозів як практичний стандарт
У задачах масового прогнозування попиту для тисяч рядів ефективними часто є ансамблі (комбінації статистичних і ML-підходів). Це підтверджено результатами M5 Accuracy Competition (ієрархічний роздрібний набір даних, близький до задач прогнозування попиту), де представлені підходи системно порівнювались на великому масиві часових рядів [14].

2. Ієрархічне прогнозування та узгодження прогнозів між рівнями
Для електронної комерції критично мати несуперечливі прогнози між рівнями (позиція → категорія → загальний попит; країна → регіон → місто). Для цього застосовують підходи узгодження прогнозів (forecast reconciliation), зокрема методи оптимального комбінування в ієрархіях і мінімізації дисперсії помилки (MinT та його модифікації) [15].

3. Побудова ознак як системна частина якості
У промислових системах активно використовують лагові ознаки, ковзні агрегати, календарні ефекти, промо та ціни, ознаки наявності запасів, а також поведінкові сигнали (перегляди, пошукові запити, додавання в кошик). Такий підхід узгоджується з логікою

роздрібних наборів рівня M5, де надані продажі разом із цінами та промо-активністю, що підкреслює роль екзогенних змінних у прогнозуванні попиту.

Для побудови прогнозних модулів дослідники та інженери широко застосовують відкриті фреймворки, що підтримують як класичні, так і сучасні моделі:

– Nixtla StatsForecast (високопродуктивні реалізації ARIMA/ETS тощо);

– Nixtla NeuralForecast (набір нейромережевих моделей, включно з трансформерами) [16,17];

– GluonTS (ймовірнісне прогнозування часових рядів);

– Darts (єдиний API для багатьох моделей + бек-тестинг і робота з екзогенними змінними);

– Kats (Meta/Facebook Research) (аналіз часових рядів, прогнозування, виявлення аномалій та ознаки) [18].

Такі інструменти дозволяють формувати прототипи та промислові рішення з контрольованим життєвим циклом і прозорими процедурами тестування.

Аналітичні системи прогнозування попиту в електронної комерції доцільно розглядати як багаторівневі рішення, де якість забезпечується не лише вибором алгоритму, а й архітектурою даних, автоматизованими конвеєрами, моніторингом і узгодженістю прогнозів у бізнес-ієрархіях. Результати досліджень і практичні корпоративні кейси підтверджують ефективність ансамблювання, ієрархічного узгодження та MLOps-автоматизації як ключових компонентів сучасних прогнозних платформ.

Висновки до розділу 1

У першому розділі показано, що попит в електронній комерції формується в умовах високої мінливості та чутливості до короткострокових стимулів. Встановлено, що для забезпечення достовірного прогнозування ключовим стає не лише вибір

алгоритму, а й якісне збирання даних, їх очищення та коректна побудова ознак, які відображають календарні ефекти, промо-впливи, цінові чинники та зовнішній контекст.

Узагальнено класи методів і критерії їх вибору. Обґрунтовано, що універсальної моделі для всіх типів попиту не існує.

Визначено, що сучасні аналітичні системи прогнозування попиту еволюціонують до багаторівневих платформ, у яких якість прогнозів підтримується не лише алгоритмічно, а й за рахунок архітектури даних, автоматизованих конвеєрів, MLOps-процесів, моніторингу деградації та узгодження прогнозів між рівнями бізнес-ієрархій.

2 ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ АНАЛІТИЧНОЇ СИСТЕМИ

2.1 Архітектурна модель та функціональні компоненти системи

Архітектура інтелектуальної аналітичної системи для прогнозування попиту в електронної комерції сформована як послідовна багаторівнева структура, яка охоплює повний життєвий цикл даних від моменту їх надходження з різних джерел до отримання готових прогнозів і аналітичних висновків. Такий підхід забезпечує чітке розмежування функцій між компонентами, можливість незалежного масштабування окремих рівнів та стабільність роботи системи за умов зростання обсягів даних і навантаження. Система підтримує пакетне оновлення моделей та онлайн-отримання прогнозів через REST API. У межах цієї архітектури виділено п’ять взаємопов’язаних рівнів (рис. 2.1).

Рівень 1. Джерела даних.

Перший рівень архітектури відповідає за надходження початкової інформації, необхідної для аналізу попиту. Саме тут система отримує транзакційні дані про продажі, інформацію про товари (атрибути, ціни, категорії), відомості про клієнтів, маркетингові активності (знижки, промоакції), зовнішні фактори (свята, погода, економічні зміни), а також вебаналітику, що відображає поведінкові події користувачів. Дані надходять у систему різними каналами, через API, пакетні завантаження або подієві канали і формують початковий потік інформації для подальших етапів обробки.

Рівень 2. Збір та інтеграція даних (ETL).

Другий рівень реалізує процеси збору та інтеграції даних відповідно до ETL-підходу. На цьому етапі виконується екстракція, очищення, трансформація, нормалізація та валідація даних, після чого вони завантажуються до відповідних сховищ. Результатом роботи цього рівня є узгоджене, очищене та структуроване середовище даних, готове до подальшого використання у моделях прогнозування.

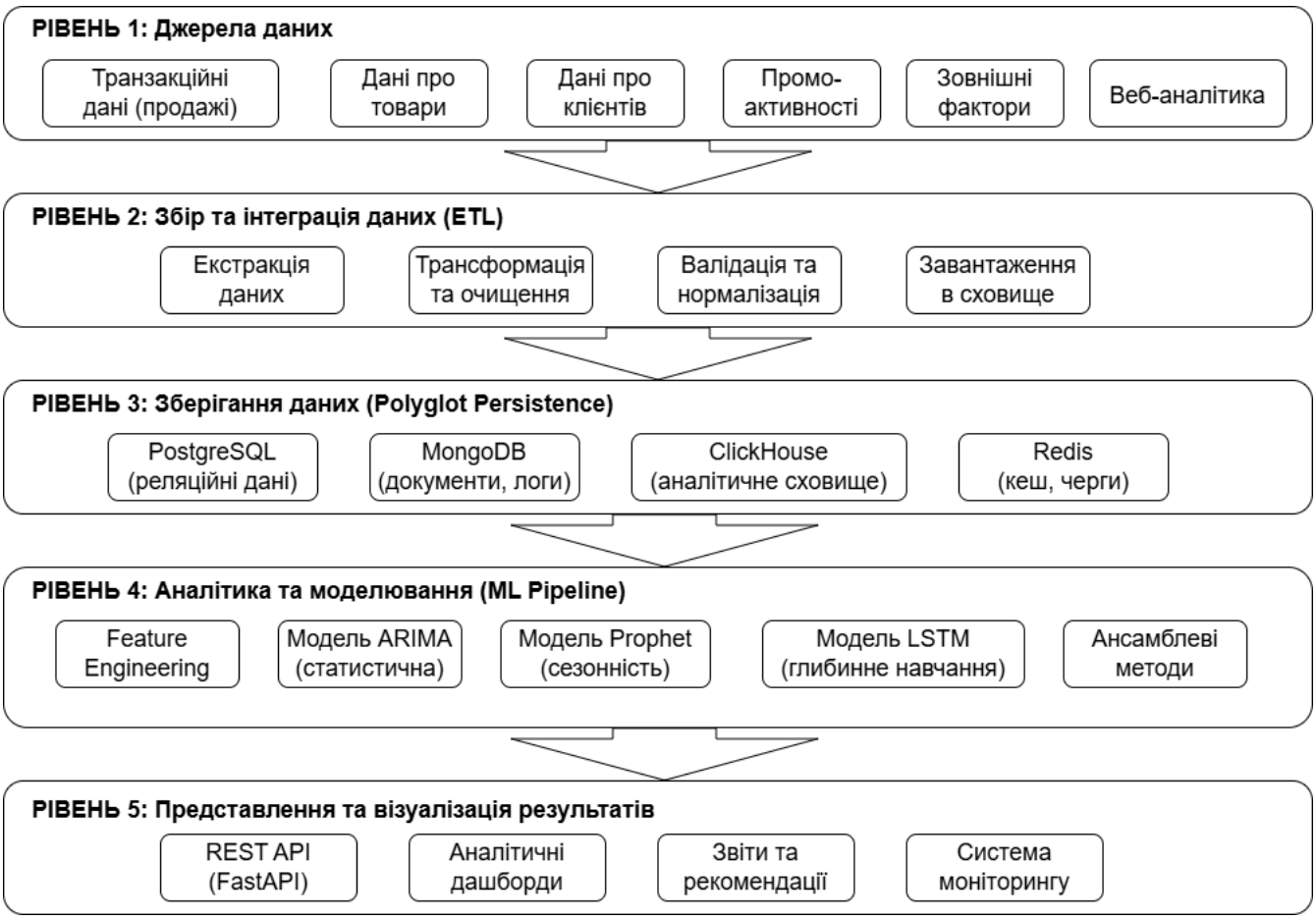


Рисунок 2.1 – Багаторівнева архітектура інтелектуальної аналітичної системи

Рівень 3. Зберігання даних (Polyglot Persistence).

Третій рівень ґрунтується на концепції Polyglot Persistence і поєднує кілька типів сховищ, кожне з яких оптимізоване для певних видів інформації [19]. PostgreSQL забезпечує надійне зберігання структурованих транзакційних даних, MongoDB використовується для документів, поведінкових логів та напівструктурованої інформації, ClickHouse обробляє великі обсяги аналітичних даних і часових рядів, тоді як Redis застосовується для кешування, швидкого доступу та роботи з чергами.

Рівень 4. Аналітика та моделювання (ML Pipeline).

Четвертий рівень присвячений аналітиці та моделюванню і включає формування ознак, навчання та оновлення моделей машинного навчання. У системі використовуються статистичні методи (ARIMA), моделі з урахуванням сезонності

(Prophet), алгоритми глибинного навчання (LSTM), а також ансамблеві техніки, що дозволяють комбінувати прогнози для підвищення їх точності та стабільності [20].

Рівень 5. Представлення та візуалізація результатів.

П’ятий рівень відповідає за представлення результатів кінцевим користувачам і зовнішнім системам. Прогнози передаються через REST API, візуалізуються у вигляді інтерактивних дашбордів і автоматичних звітів, а також супроводжуються моніторингом якості моделей, що дозволяє своєчасно реагувати на зміни у поведінці попиту.

Для ілюстрації динамічної взаємодії компонентів системи створено типовий сценарій отримання прогнозу попиту на товар, представлений на діаграмі послідовності (рис. 2.2).

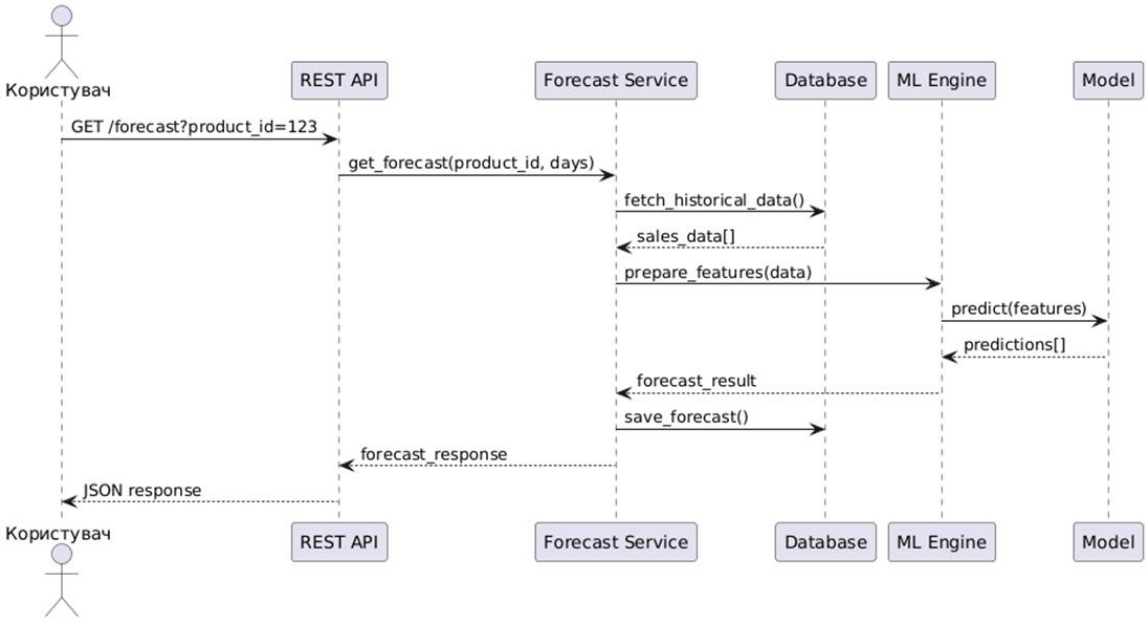


Рисунок 2.2 – Діаграма послідовностей

Процес починається з ініціації запиту користувачем через REST API із зазначенням ідентифікатора товару та горизонту прогнозування. API передає запит до Forecast Service, який координує всі подальші операції через звернення до Database Manager для отримання історичних даних, передачу їх до ML Engine для підготовки ознак та генерації прогнозу відповідною моделлю. Після генерації прогнозу Forecast

Service зберігає результат у базі даних та формує відповідь з прогнозними значеннями, довірчими інтервалами та метриками достовірності, яку REST API повертає користувачу у форматі JSON.

Така послідовність операцій демонструє чітке розділення відповідальності між компонентами та забезпечує модульність системи. Кожен компонент може бути замінений або модифікований без впливу на інші частини системи, що є важливою характеристикою з точки зору підтримки та розвитку.

2.2 Організація даних і застосування концепції Polyglot Persistence

Ефективна організація даних є ключовою передумовою високої продуктивності та надійності системи прогнозування попиту. Різнорідність даних, що включають транзакційні записи, лог-файли, часові ряди та результати експериментів, не дозволяє ефективно використовувати одне універсальне сховище. Тому в архітектурі системи реалізовано підхід Polyglot Persistence, який передбачає застосування кількох спеціалізованих сховищ, оптимізованих під конкретний тип навантаження, структуру даних та сценарії доступу [19, 21]. Жодна окрема технологія баз даних, будь то SQL чи NoSQL, не може задовольнити всі потреби бізнесу та вирішити всі технологічні проблеми. Щоб вибрати правильну базу даних, необхідно враховувати набір критеріїв: модель даних, підтримку CAP, ємність, продуктивність, API запитів, надійність, стійкість даних, перебалансування та підтримку бізнесу [22].

1. Концепція Polyglot Persistence.

Традиційний підхід, що передбачає використання однієї СКБД для всіх типів даних, стає неефективним у сучасних аналітичних системах з високими вимогами до масштабованості, гнучкості та швидкодії. Polyglot Persistence пропонує архітектурне рішення (рис. 2.3), яке полягає у поєднанні кількох баз даних, кожна з яких виконує чітко визначене завдання.

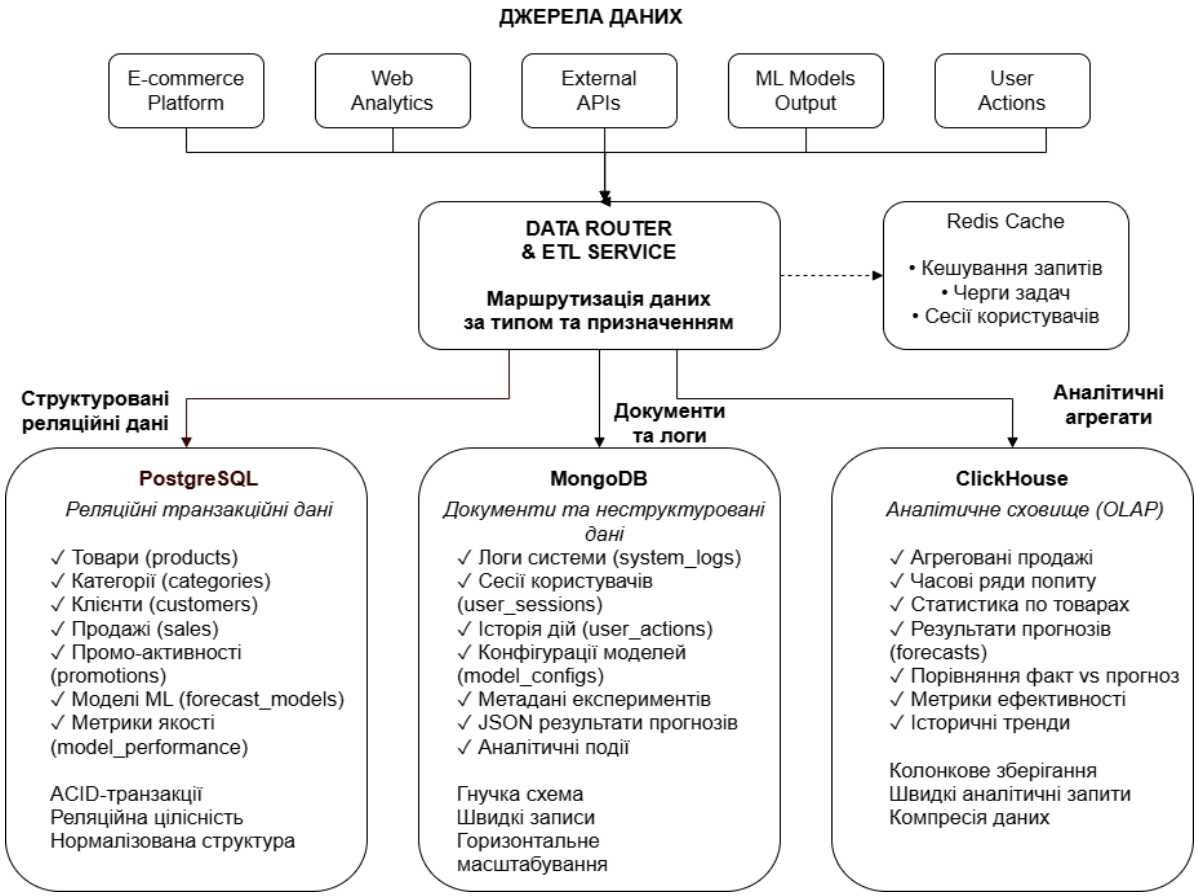


Рисунок 2.3 - Polyglot Persistence

У межах системи прогнозування попиту дані поділяються на кілька груп, кожна з яких має відмінні характеристики:

– структуровані транзакційні дані (товари, продажі, клієнти) – потребують ACID-транзакцій, строгих зв’язків та складних SQL-запитів;

– напівструктуровані дані (логи, конфігурації моделей, результати експериментів) – вимагають гнучкої схеми та швидкого запису;

– аналітичні агрегати (часові ряди, статистичні показники, контекстні ознаки) – потребують можливості виконання складних OLAP-запитів над великими обсягами даних;

– тимчасові дані й кеш (проміжні обчислення, результати прогнозів, черги задач) – потребують мінімальної затримки доступу та керованого TTL.

Маршрутизація даних між цими сховищами здійснюється через компонент Data Router & ETL Service, що визначає призначення даних, виконує їх трансформацію та координує синхронізацію між сховищами.

Переваги підходу Polyglot Persistence для цієї системи полягають у:

– підвищенні продуктивності за рахунок спеціалізації сховищ;

– незалежне масштабування компонентів, що дозволяє збільшувати ресурси лише для тих сховищ, які відчувають найбільше навантаження;

– підвищенні стійкості за рахунок розподілу навантаження;

– гнучкість архітектури, можливість інтеграції нових інструментів без повної перебудови.

Застосування підходу Polyglot Persistence підвищує продуктивність системи за рахунок спеціалізації сховищ під різні типи навантаження (OLTP/OLAP/логи/кеш), забезпечує незалежне масштабування окремих компонентів та підвищує стійкість системи завдяки розподілу даних і сервісів між кількома сховищами [23] . Крім того, архітектура зберігає гнучкість щодо заміни або додавання технологій у разі зміни вимог.

2. Реляційна база даних PostgreSQL (https://www.postgresql.org/).

PostgreSQL є основним сховищем структурованих транзакційних даних завдяки підтримці ACID, широким можливостям SQL, високій надійності та механізмам індексування. Реляційна модель використовується для зберігання довідників, операційних даних та метаданих моделей.

Структура реляційної БД подана на ER-діаграмі.

Основні групи таблиць:

Довідники:

– categories – ієрархія категорій (category_id, parent_category_id, category_name, description).

– products – центральний довідник товарів, що включає характеристики, ціни, статус активності та часові мітки (product_id, name, category_id, price, cost, stock_quantity, dimensions тощо).

– customers – профілі клієнтів із сегментацією та агрегованими показниками активності (customer_id, email, segment, total_spent, geography тощо).

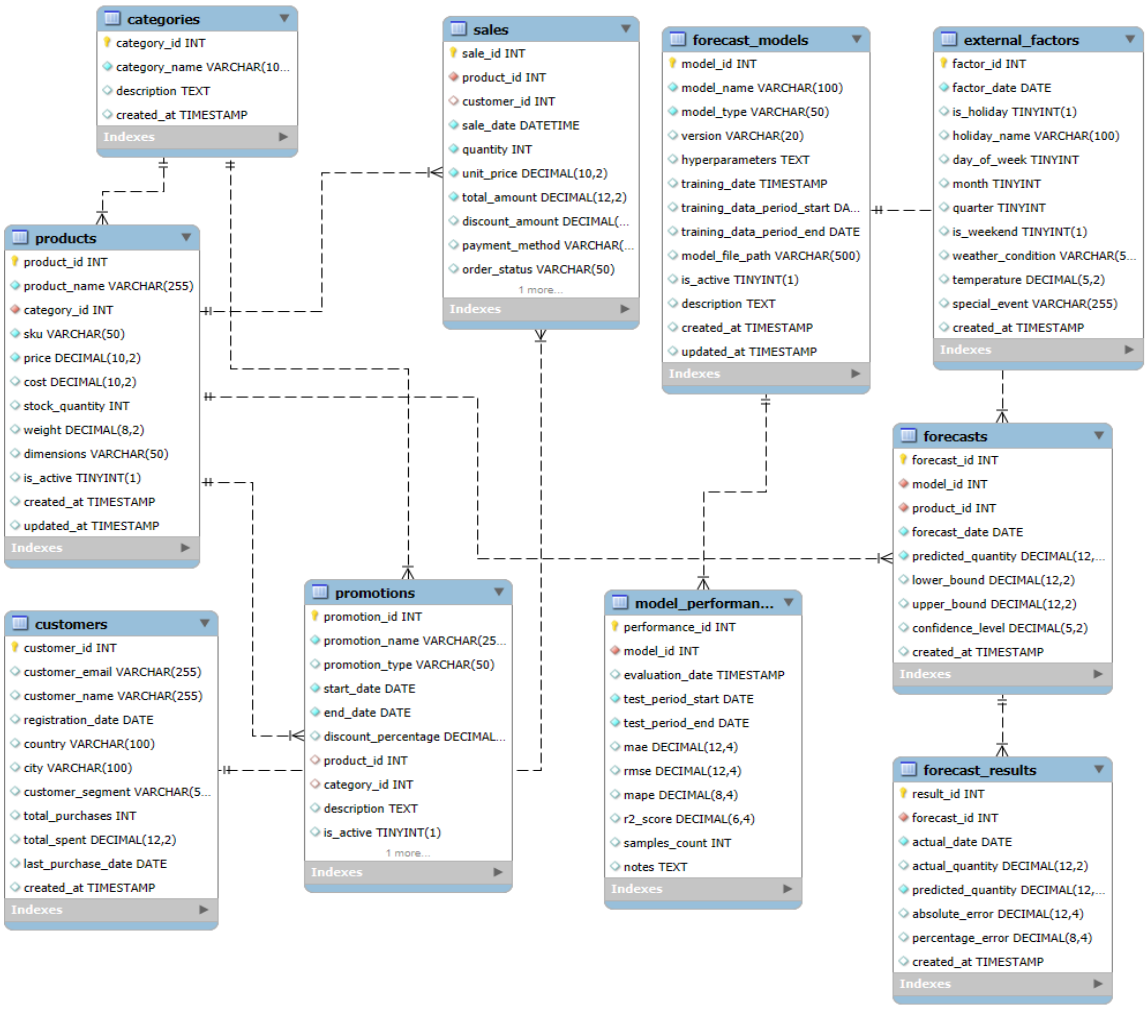


Рисунок 2.4 – ER-діаграма структури БД

Операційні таблиці:

– sales – журнал усіх продажів з повною транзакційною інформацією. Для прискорення аналітичних запитів реалізовано індекси за product_id, sale_date, customer_id та композитний індекс (product_id, sale_date).

– promotions – інформація про промо-акції (promotion_type, discount, action period).

– external_factors – зовнішні чинники та календарні атрибути, що використовуються при формуванні ознак: свята, погода, сезонність, події. Зв’язування з продажами виконується логічним об’єднанням за датою під час підготовки датасету ознак (наприклад, factor_date ↔ sale_date::date), без обов’язкового зовнішнього ключа на рівні схеми БД.

Таблиці машинного навчання

– forecast_models – збереження моделей, гіперпараметрів та метаданих навчання;

– forecasts – прогнози по товарах із довірчими інтервалами;

– model_performance – метрики якості моделей (MAE, RMSE, MAPE, R^2);

– forecast_results – порівняння прогнозів з фактичними даними.

Цілісність моделі підтримується через зовнішні ключі із правилами каскадування (ON DELETE CASCADE / SET NULL), що дає змогу зберігати історичні дані навіть при зміні довідників.

3. Документноорієнтована БД MongoDB (https://www.mongodb.com/)

MongoDB застосовується для зберігання напівструктурованих та швидкомінливих даних, які не підходять для жорсткої реляційної схеми. Перевагами використання MongoDB є гнучка схема без потреби у міграціях, висока швидкість вставки даних, критична для логування в реальному часі, природна підтримка вкладених структур, можливість горизонтального масштабування (sharding) [24].

Основні колекції:

– system_logs – детальні події роботи системи з можливістю довільного розширення структури (level, component, message, context, trace_id);

– user_sessions – сесії користувачів із масивом дій та метаданими;

– model_configs – конфігурації моделей у форматі JSON, включно з гіперпараметрами та правилами feature engineering;

– experiment_results – результати експериментів: метрики, опис датасетів, посилання на графіки та примітки дослідників.

4. Аналітичне сховище ClickHouse (https://clickhouse.com/).

ClickHouse забезпечує виконання високопродуктивних аналітичних запитів та використовується як OLAP-сховище для великих агрегатів [25].

Основні таблиці:

– daily_sales_aggregates – денні показники продажів (quantity, revenue, unique_customers тощо);

– forecast_time_series – часові ряди прогнозів для швидкої аналітики;

– model_performance_metrics – історія метрик моделей у динаміці;

– demand_patterns – закономірності попиту (тренди, сезонність, варіація).

ClickHouse демонструє високу швидкодію завдяки колонковому зберіганню, компресії, векторизованому виконанню та можливості горизонтального масштабування.

5. Кеш Redis.

Redis використовується для високошвидкісного доступу до даних та асинхронної обробки задач.

Основні випадки використання:

– кешування результатів прогнозів з ключами типу "forecast:{product_id}:{date}" для швидкого доступу без звернення до основної бази даних. Час життя (TTL) встановлюється залежно від горизонту прогнозування;

– кешування метаданих товарів та категорій дозволяє уникнути частих запитів до PostgreSQL при формуванні ознак для прогнозування;

– черги задач для асинхронної обробки реалізуються через структури даних Redis Lists. Задачі на перенавчання моделей, генерацію прогнозів та формування звітів поміщаються в черги та обробляються воркерами;

– зберігання сесій користувачів забезпечує швидкий доступ до даних активних сесій без навантаження на основну базу даних;

– розподілені блокування (distributed locks) використовуються для координації роботи різних екземплярів сервісів та запобігання одночасному виконанню критичних операцій.

Redis налаштований з політикою видалення застарілих ключів (eviction policy) типу "allkeys-lru" для автоматичного звільнення пам'яті при досягненні ліміту. Реплікація реплікація primary–replica (master–replica) забезпечує надійність даних кешу [26].

У системах з Polyglot Persistence особливо важливо забезпечити узгодженість даних між різними сховищами. У системі реалізовано комбіновану стратегію синхронізації:

– ETL-синхронізація за розкладом: дані з PostgreSQL агрегуються та завантажуються до ClickHouse кожні 5–15 хвилин (критичні) або щогодини/добу (аналітичні).

– подійно-орієнтована синхронізація: після запису нового прогнозу генерується подія, що оновлює ClickHouse і інвалідовує відповідні ключі Redis.

– Cache-Aside патерн: при відсутності значення в Redis дані беруться з PostgreSQL і кешуються; при оновленні – кеш очищається.

– Saga патерн використовується для забезпечення атомарності складних процесів (збереження прогнозу, оновлення аналітики, інвалідація кешу).

– моніторинг консистентності: регулярна перевірка відповідності агрегатів у ClickHouse вихідним даним у PostgreSQL та контроль затримок реплікації.

Така архітектура забезпечує:

– високу продуктивність завдяки спеціалізації кожного сховища;

– масштабованість через незалежне збільшення ресурсів для окремих компонентів;

– відмовостійкість, оскільки збої одного сховища не призводять до недоступності системи загалом;

– гнучкість розвитку, що дозволяє легко інтегрувати нові технології або замінювати окремі компоненти;

– стійку основу для роботи моделей прогнозування, оскільки дані зберігаються у структурі, яка повністю відповідає вимогам ML-пайплайнів.

Таким чином, реалізований підхід Polyglot Persistence забезпечує оптимальний баланс між продуктивністю, надійністю та масштабованістю, створюючи ефективне та розширюване середовище для системи прогнозування попиту..

2.3 Моделі прогнозування попиту та алгоритмічні рішення

Ядром системи прогнозування попиту є набір моделей машинного навчання, що реалізують різні підходи до аналізу часових рядів. Використання декількох моделей з різними математичними основами дозволяє врахувати різноманітні характеристики попиту та забезпечити стабільність прогнозів.

Процес прогнозування попиту організовано у вигляді ML Pipeline – послідовності етапів від збору вхідних даних до генерації та збереження прогнозів. Детальна блок-схема цього процесу представлена у Додатку А, рис. А.1.

ML Pipeline складається з шести основних етапів:

1. Збір та підготовка даних – завантаження історичних даних з БД, очищення від дублікатів та викидів, валідація типів даних, агрегація транзакцій до денного рівня.

2. Feature Engineering – створення інформативних ознак для моделей (часові, лагові, ковзні середні, сезонні компоненти).

3. Навчання моделей паралельне навчання ARIMA, Prophet, LSTM, XGBoost з оптимізацією гіперпараметрів.

4. Оцінка якості включає валідацію кожної моделі на тестових даних, розрахунок метрик якості (MAE, RMSE, MAPE, R^2), порівняння моделей та вибір найкращої.

5. Прогнозування передбачає генерацію прогнозів на майбутній період за допомогою обраної або ансамблевої моделі, розрахунок довірчих інтервалів для оцінки невизначеності прогнозу, збереження навченої моделі у файловому сховищі, запис результатів прогнозів у базу даних. Артефакти моделі зберігаються у файловому сховищі, тоді як метадані та метрики – у PostgreSQL/ClickHouse.

6. Моніторинг та оновлення забезпечує постійний контроль якості прогнозів через порівняння з фактичними значеннями, виявлення деградації якості моделі з часом, автоматичне перенавчання моделей при погіршенні метрик нижче порогових значень або за розкладом.

Якість прогнозів істотно залежить від інформативності ознак, що подаються на вхід моделей. У системі реалізовано комплексний підхід до створення ознак декількох категорій (табл. 2.1).

Таблиця 2.1 – Категорії ознак для прогнозування попиту

Категорія ознак	Опис та приклади
Часові ознаки	Рік, місяць, день тижня, квартал, is_weekend, is_holiday. Циклічне кодування: sin/cos для місяців
Лагові ознаки	Продажі 1, 7, 14, 30 днів тому; продажі в той самий день минулого місяця/року
Ковзні середні	MA(7), MA(14), MA(30); EMA; ковзне стандартне відхилення; min/max по вікну
Трендові ознаки	Лінійний тренд (коефіцієнт нахилу); відношення до MA; різниця між MA різних періодів
Сезонні компоненти	STL-декомпозиція (тренд, сезонність, залишки); індекси сезонності для місяців та днів тижня
Ознаки товару	Категорія (One-Hot/Target Encoding); ціна та її зміни; наявність на складі; характеристики товару
Промо-активності	is_promotion; discount_percentage; тип промо-акції; кількість днів від початку/до кінця промо
Зовнішні фактори	is_holiday; назва свята; погодні умови (temperature, weather_condition); спеціальні події
Агрегати по категоріях	Середній/загальний попит по категорії; частка товару

Нормалізація ознак виконується для забезпечення однакового масштабу. Застосовується StandardScaler (стандартизація до середнього 0 та дисперсії 1) або MinMaxScaler (нормалізація до діапазону [0, 1]) залежно від типу моделі. Для лінійних моделей та нейронних мереж нормалізація є критично важливою для швидкої збіжності навчання.

ARIMA (AutoRegressive Integrated Moving Average) є класичною статистичною моделлю для аналізу та прогнозування часових рядів [20, 21]. Модель комбінує три компоненти: авторегресійну (AR), інтегровану (I) та ковзного середнього (MA).

Повна модель ARIMA(p, d, q) у термінах лагового оператора B B задається рівнянням:

$$\varphi(B)(1-B)^d y_t = \theta(B)\varepsilon_t \quad (2.1)$$

де $\varphi(B) = 1 - \varphi_1 B - \cdots - \varphi_p B^p$ – поліном авторегресії порядку p,

$\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$ – поліном ковзного середнього порядку q,

d– порядок диференціювання,

B– оператор зсуву назад ($By_t = y_{t-1}$),

ε_t– білий шум.

Для врахування сезонності використовується розширення SARIMA(p,d,q)(P,D,Q,s), де s – період сезонності (7 для тижневої, 365 для річної) [22].

Параметри p, d, q підбираються автоматично за допомогою мінімізації інформаційних критеріїв AIC або BIC, або аналізом автокореляційної (ACF) та часткової автокореляційної (PACF) функцій.

Переваги ARIMA: математична обґрунтованість, інтерпретованість коефіцієнтів, добра робота на стаціонарних рядах, швидке навчання.

Недоліки: обмежена здатність моделювати нелінійні залежності, чутливість до викидів, складність врахування зовнішніх факторів.

У системі ARIMA використовується як базова модель для товарів зі стабільним попитом. Реалізація виконана за допомогою бібліотеки pmdarima (auto_arima) [23].

Модель Prophet: врахування сезонності та трендів.

Prophet – модель прогнозування часових рядів, розроблена Facebook (Meta), яка особливо добре підходить для даних з виразною сезонністю, трендами та пропущеними значеннями [24].

Модель представляє часовий ряд як суму компонент:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t, \quad (2.2)$$

де g(t) – функція тренду (кусково-лінійна або логістична),

s(t) – сезонна компонента (ряди Фур'є),

h(t) – вплив святкових днів,

ε_t – помилка.

Сезонна компонента моделюється за допомогою рядів Фур'є:

$$s(t) = \sum_{i=1}^{N} \left(a_i cos\left(\frac{2\pi nt}{P}\right) + b_i sin\left(\frac{2\pi it}{P}\right)\right) \quad (2.3)$$

де P– період сезонності (наприклад, 7для тижневої та 365.25для річної),

i– номер гармоніки,

a_i і b_i– коефіцієнти, що оцінюються з даних,

N– кількість гармонік (порядок ряду).

Prophet автоматично виявляє точки зламу тренду (changepoints) та дозволяє явно задати список святкових днів з оцінкою їх впливу на основі історичних даних.

Основні гіперпараметри: changepoint_prior_scale (гнучкість тренду), seasonality_prior_scale (сила сезонності), holidays_prior_scale (вплив святкових днів), fourier_order (порядок рядів Фур'є).

Переваги Prophet: автоматичне виявлення сезонності та трендів, природна обробка пропусків, явне моделювання святкових днів, інтуїтивні гіперпараметри, швидке навчання, автоматичні довірчі інтервали.

Недоліки: обмежена здатність моделювати складні нелінійні взаємодії, менша гнучкість порівняно з deep learning.

У системі Prophet використовується для товарів з виразною сезонністю та для випадків, коли важливо враховувати вплив конкретних святкових днів [25].

Модель LSTM (Long Short-Term Memory) – тип рекурентної нейронної мережі, спеціально розроблений для роботи з послідовними даними та здатний запам'ятовувати довгострокові залежності [20, 22].

Архітектура LSTM вирішує проблему зникаючого градієнта за допомогою спеціальної структури осередку з трьома вентилями: вентиль забування (forget gate)

визначає, яку інформацію з попереднього стану слід забути; вхідний вентиль (input gate) визначає, яку нову інформацію додати; вихідний вентиль (output gate) контролює, яка частина стану буде виведена.

Архітектура мережі для прогнозування попиту:

– вхідний шар: послідовність векторів ознак довжиною lookback (наприклад, 30 днів історії) ;

– LSTM шари (1-3 шари з 50-200 нейронами): обробка послідовності, виділення темпоральних патернів ;

– Dropout шари (0.2-0.3): регуляризація для запобігання перенавчанню;

– Dense шари: фінальне перетворення виходу LSTM ;

– вихідний шар: прогноз на horizon кроків вперед (наприклад, 7 днів) .

Підготовка даних: з часового ряду формуються вікна фіксованої довжини lookback для прогнозування наступних horizon значень. Дані обов'язково нормалізуються за допомогою MinMaxScaler або StandardScaler.

Навчання включає ініціалізацію архітектури, компіляцію моделі з функцією втрат (MSE або MAE) та оптимізатором (Adam), навчання з early stopping для запобігання перенавчанню.

Основні гіперпараметри: кількість LSTM шарів та нейронів, lookback period, horizon, dropout rate, batch size, learning rate, epochs.

Переваги LSTM: здатність моделювати складні нелінійні залежності, природна обробка послідовностей, запам'ятовування довгострокових залежностей, гнучкість архітектури. Недоліки: потреба у великих обсягах даних (мінімум кілька тисяч спостережень), тривалий час навчання, складність інтерпретації, чутливість до гіперпараметрів, ризик перенавчання.

У системі LSTM використовується для товарів з складними нелінійними патернами та достатньою історією даних (мінімум 1-2 роки щоденних спостережень). Реалізація виконана за допомогою TensorFlow/Keras [28].

Модель XGBoost (eXtreme Gradient Boosting) належить до класу алгоритмів gradient boosting, що послідовно будують ансамбль слабких моделей (decision trees), кожна з яких прагне виправити помилки попередніх [27].

Для адаптації XGBoost до задачі прогнозування часових рядів застосовується перетворення у задачу supervised learning: створюються лагові ознаки, цільова змінна – значення попиту. XGBoost навчається на таблиці "ознаки → цільова змінна" та будує нелінійну залежність між ними.

Основні гіперпараметри: n_estimators (кількість дерев, 100-500), max_depth (глибина дерев, 3-10), learning_rate (швидкість навчання, 0.01-0.3), subsample (частка даних для кожного дерева, 0.8-1.0), colsample_bytree (частка ознак, 0.8-1.0), регуляризаційні параметри alpha та lambda.

Переваги XGBoost: висока точність завдяки ансамблевому підходу, автоматичне виявлення нелінійних взаємодій, робастність до викидів та пропусків, можливість аналізу важливості ознак (feature importance), ефективна реалізація з паралелізацією.

Ансамблевий підхід комбінує прогнози різних моделей для підвищення стабільності. У системі реалізовано два методи:

Weighted Average (зважене усереднення):

$$\hat{y}_{\text{ens}} = \sum_{i=1}^{4} w_i \hat{y}_i \,, \sum_{i=1}^{4} w_i = 1,\ w_i \geq 0, \tag{2.4}$$

де ваги обчислюються як

$$w_i = \frac{1/\text{error}_i}{\sum_{j=1}^{4} (1/\text{error}_j)}. \tag{2.5}$$

Stacking (мета-навчання): прогнози базових моделей використовуються як ознаки для навчання мета-моделі (лінійної регресії або легкого gradient boosting), що навчається на валідаційній вибірці для запобігання перенавчанню. У системі XGBoost використовується як окрема модель для товарів з багатьма ознаками та складними залежностями. Ансамблевий підхід застосовується для критичних товарів, де потрібна максимальна точність [30].

Метрики оцінки якості прогнозів.

Для оцінки якості моделей використовується набір метрик.

MAE – середня абсолютна помилка.

RMSE – середньоквадратична помилка.

MAPE – середня відсоткова помилка.

R^2 – частка дисперсії.

При оцінці моделей використовується time series cross-validation з розділенням даних на тренувальний (70%), валідаційний (15%) та тестовий (15%) набори зі збереженням часового порядку. Модель оцінюється на декількох тестових вікнах для перевірки стабільності якості.

Вибір найкращої моделі або створення ансамблю здійснюється на основі комплексного аналізу метрик на валідаційному наборі, стабільності метрик на різних тестових вікнах, обчислювальної ефективності та інтерпретованості результатів. Для різних категорій товарів можуть бути оптимальними різні моделі або їх комбінації. Таким чином, підхід до прогнозування попиту з використанням різноманітних моделей машинного навчання забезпечує точність та надійність прогнозів для різних типів товарів. Гнучка архітектура ML Pipeline дозволяє легко додавати нові моделі та експериментувати з різними комбінаціями підходів.

Висновки до розділу 2

У розділі 2 сформовано архітектурну основу інтелектуальної аналітичної системи прогнозування попиту в електронної комерції та обґрунтовано вибір ключових технологічних рішень. Архітектуру подано як багаторівневу структуру, що охоплює повний цикл даних від надходження з різних джерел до отримання прогнозів і їх представлення користувачам через API та засоби візуалізації. Запропоноване розмежування на рівні (джерела даних, ETL-інтеграція, зберігання, ML-аналітика, представлення результатів) забезпечує модульність, спрощує супровід і дає можливість незалежного масштабування компонентів відповідно до навантаження.

Показано, що різнорідність даних і сценаріїв доступу зумовлює застосування підходу Polyglot Persistence, у межах якого використано спеціалізовані сховища: Для забезпечення узгодженості між компонентами визначено комбіновану стратегію синхронізації, що включає ETL-оновлення за розкладом, подійно-орієнтовані механізми, патерн Cache-Aside та Saga-підхід для багатокрокових операцій із компенсуючими діями, а також моніторинг консистентності агрегатів.

Окрему увагу приділено процесу прогнозування у вигляді ML Pipeline, який охоплює підготовку даних, побудову ознак, навчання й оцінювання моделей, генерацію прогнозів та моніторинг якості з можливістю регулярного оновлення. Обґрунтовано використання набору моделей із різними математичними підходами, а також ансамблевих стратегій, що підвищує стійкість і точність прогнозування для різних типів товарів і патернів попиту.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Експериментальне середовище та характеристики даних

Для проведення експериментальних досліджень та оцінки ефективності розробленої інтелектуальної системи прогнозування попиту використано реальні дані роздрібної торгівлі, що дозволяє об'єктивно оцінити якість різних підходів до прогнозування та обґрунтувати прийняті архітектурні рішення.

Для експериментальних досліджень обрано датасет Walmart Store Sales Forecasting, який містить реальні дані про продажі однієї з найбільших роздрібних мереж США [28]. Датасет широко використовується в академічних дослідженнях та промисловості як benchmark датасет для оцінки алгоритмів прогнозування попиту [29]

Вибір саме цього датасету обумовлений кількома факторами. По-перше, це реальні дані великої роздрібної компанії, що забезпечує практичну релевантність досліджень. По-друге, достатній обсяг історичних даних (майже 3 роки) дозволяє навчити складні моделі машинного навчання та виявити сезонні патерни. По-третє, наявність зовнішніх факторів (погода, економічні показники, свята) дозволяє оцінити вплив контекстуальної інформації на якість прогнозів. По-четверте, датасет є загальновизнаним benchmark датасетом, що дозволяє порівняти результати з іншими дослідженнями. По-п'яте, різноманітність департаментів з різними характеристиками попиту (стабільний, сезонний, волатильний) дозволяє оцінити універсальність підходів.

Датасет складається з трьох основних компонентів.

Файл stores.csv містить інформацію про 45 магазинів мережі Walmart, розташованих у різних регіонах США, з характеристиками типу магазину (Type A, B, C) та розміру (Size у квадратних футах).

Файл train.csv містить історичні дані про тижневі продажі по 99 департаментах кожного магазину за період з лютого 2010 року по жовтень 2012 року, загалом 421 570 записів. Кожен запис містить ідентифікатор магазину (Store), номер департаменту

(Dept), дату (Date), тижневі продажі (Weekly_Sales) та прапорець святкового тижня (IsHoliday).

Файл features.csv містить зовнішні фактори для кожного магазину та тижня: температуру (Temperature), вартість палива (Fuel_Price), індекс споживчих цін (CPI), рівень безробіття (Unemployment), наявність промо-акцій різних типів (MarkDown1-5).

Структура датасету наведена на рисунку 3.1.

ТАБЛИЦЯ 3.1 - Структура датасету Walmart

Файл	Кількість записів	Основні поля
train.csv	421570	Store, Dept, Date, Weekly_Sales, IsHoliday
stores.csv	45	Store, Type, Size
features.csv	8190	Store, Date, Temperature, Fuel_Price, CPI, Unemployment, MarkDown1-5
Об'єднаний датасет	421570	Всі поля об'єднано

Рисунок 3.1 –Структура датасету Walmart Store Sales Forecasting

Загальний обсяг датасету становить 421 570 записів продажів, що охоплює період 143 тижні (близько 33 місяців), 45 магазинів та 99 категорій товарів (департаментів). Частота даних – тижнева, що є типовим для аналізу попиту в роздрібній торгівлі. Цільова змінна – Weekly_Sales (тижневі продажі в доларах США).

1. Дослідницький аналіз даних (Exploratory Data Analysis, EDA)

Перед побудовою моделей прогнозування проведено детальний дослідницький аналіз даних (Exploratory Data Analysis, EDA) для виявлення основних характеристик та закономірностей.

Аналіз розподілу продажів показав, що тижневі продажі по департаментах мають істотний розкид: мінімальне значення становить від'ємну величину (–4,989 доларів, що відповідає поверненням товарів), медіанне значення – 7,612 доларів, середнє значення – 15,981 долар, максимальне значення – 693,099 доларів. Розподіл продажів має правостороннє скошення (positive skew), що характерно для даних роздрібної торгівлі: більшість департаментів має помірні продажі, але існують окремі періоди з дуже високими продажами (наприклад, передсвятковий період).

Гістограма розподілу тижневих продажів наведена на рис. 3.2..

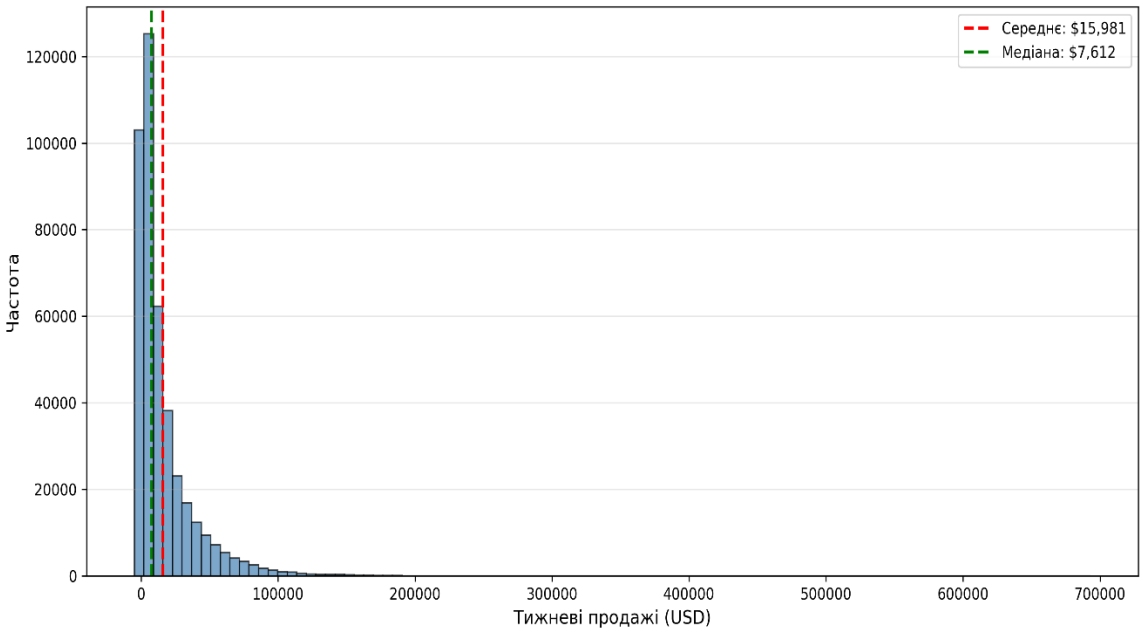


Рисунок 3.2 – Розподіл тижневих продажів по департаментах

Аналіз часових трендів показав змінну динаміку продажів протягом періоду спостереження. Середні продажі демонстрували невелике зниження: 2010 рік – $16 270,28; 2011 рік – $15 954,07 (–1,9%); 2012 рік – $15 694,95 (–1,6%). Середній річний темп зниження становив –1,7%. Проте спостерігається чітко виражена річна сезонність з піками продажів у святкові періоди: найвищі продажі спостерігаються на тижні з Днем подяки (Thanksgiving) та Різдвом (December), помітне зростання також на тижнях з Днем праці (Labor Day) та Днем незалежності (Independence Day).

Динаміка середніх тижневих продажів наведена на рисунку 3.3.

Аналіз впливу зовнішніх факторів показав наступні закономірності.

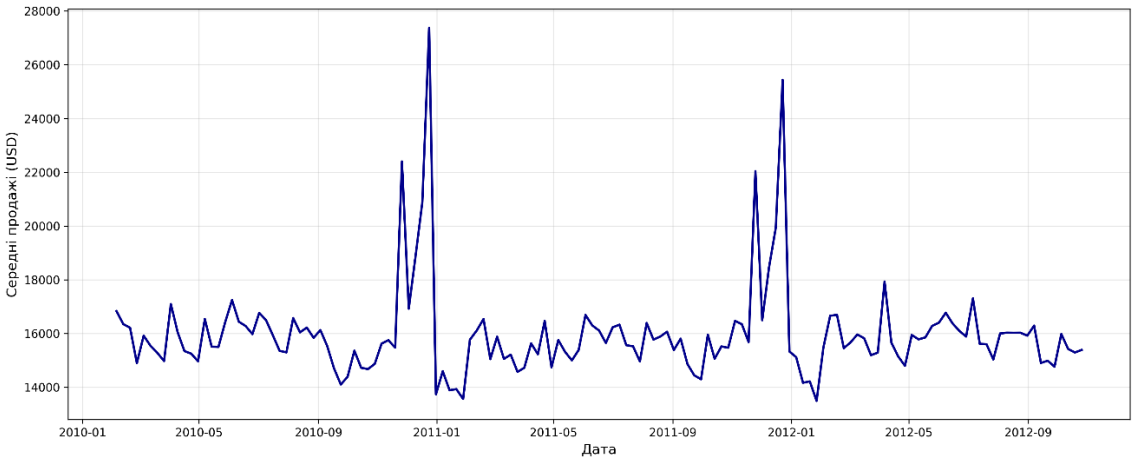


Рисунок 3.3 – Динаміка середніх продажів за період спостереження

Святкові тижні (IsHoliday = True) в середньому мають на 23,4% вищі продажі порівняно з несвятковими тижнями (несвяткові тижні: $15 792,26; святкові тижні: $19 483,45), проте існує істотна варіативність по департаментах.

Порівняння продажів у святкові та несвяткові тижні представлено на рисунку 3.4.

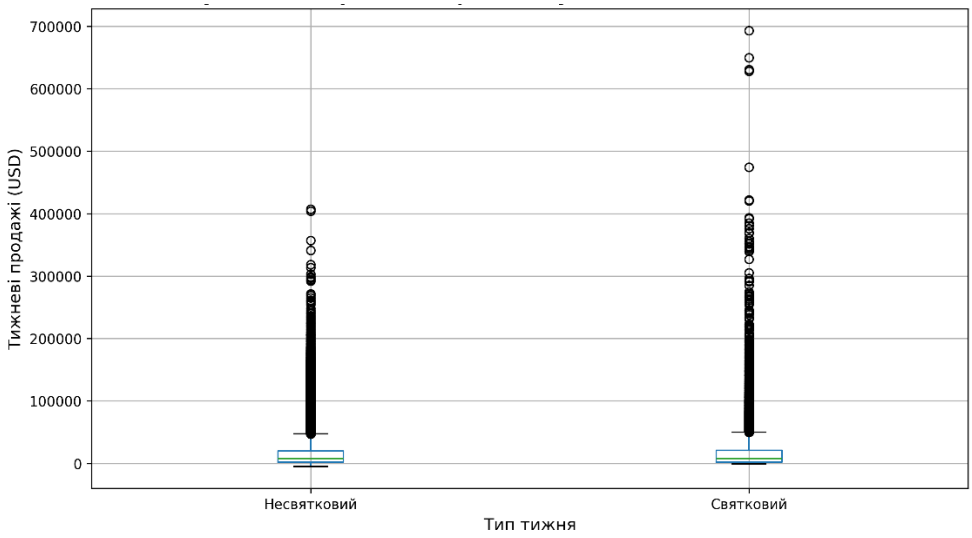


Рисунок 3.4 – Порівняння продажів у святкові та несвяткові тижні

Кореляційний аналіз зовнішніх факторів виявив наступні залежності.

Температура має слабкий позитивний вплив на продажі (r = +0,015), що пояснюється сезонною специфікою окремих категорій товарів.

Вартість палива демонструє негативну кореляцію з продажами (r = –0,028), що може відображати зменшення купівельної спроможності при зростанні витрат на пересування. Індекс споживчих цін (CPI) має слабку позитивну кореляцію (r = +0,038). Рівень безробіття демонструє слабку негативну кореляцію з продажами (r = –0,062), відображаючи загальний стан економіки. Кореляційна матриця зовнішніх факторів представлена на рисунку 3.5.

Рисунок 3.5 – Кореляційна матриця зовнішніх факторів та продажів

Аналіз різних типів департаментів за коефіцієнтом варіації (відношення стандартного відхилення до середнього значення) виявив три основні групи за характером попиту.

Стабільні департаменти (близько 33% від загальної кількості) мають відносно постійний попит з низькою волатильністю (CV < 0,75), слабкою сезонністю та передбачуваними трендами. Для таких департаментів ефективні прості статистичні моделі типу ARIMA. Сезонні департаменти (близько 33%) демонструють виражену сезонність з регулярними піками у певні періоди року, помірну волатильність (0,75 ≤ CV

< 1,5) та чутливість до святкових днів. Для них оптимальні моделі типу Prophet з явним моделюванням сезонності. Волатильні департаменти (близько 33%) характеризуються високою мінливістю попиту (CV ≥ 1,5), нерегулярними піками, сильною чутливістю до промо-акцій та складними нелінійними залежностями. Для таких департаментів найкраще підходять моделі глибинного навчання (LSTM) або gradient boosting (XGBoost).

Приклади часових рядів для різних типів департаментів наведено на рисунку 3.6.

Рисунок 3.6 – Приклади часових рядів для різних типів департаментів

2. Аналіз якості даних

Аналіз якості даних виявив відсутність пропущених значень у полі Weekly_Sales (цільова змінна), що забезпечує можливість навчання моделей без додаткової імпутації. Однак у файлі features.csv виявлено пропущені значення у полях MarkDown1-5 (близько 50-60% пропусків), що відповідає періодам без активних промо-акцій. Ці пропуски

заповнюються нульовими значеннями, що інтерпретується як відсутність промо. Декілька записів з від'ємними продажами (1 285 записів, 0,3% від загального обсягу) збережено в датасеті як є, оскільки вони відображають реальні бізнес-процеси повернення товарів. Викидів у даних виявлено небагато (менше 1% спостережень), і вони переважно відповідають реальним подіям (наприклад, екстремально високі продажі в передсвятковий період).

3. Підготовка даних для моделювання

На основі проведеного попереднього аналізу сформовано єдиний датасет для навчання та оцінки моделей шляхом об'єднання даних з трьох файлів за ключами Store та Date. Процес підготовки включав декілька етапів:

Завантаження вихідних файлів CSV у pandas DataFrame здійснено з автоматичним визначенням типів даних.

Перетворення дати виконано шляхом конвертації поля Date у тип datetime для коректної роботи з часовими рядами.

Об'єднання таблиць проведено через операцію merge: спочатку train.csv з features.csv за ключами (Store, Date), потім результат з stores.csv за ключем Store.

Обробка пропущених значень у промо-акціях (MarkDown1-5) виконана заповненням нулями, що інтерпретується як відсутність активних промо.

Створення додаткових ознак включало екстракцію часових компонентів (Year, Month, Week, DayOfYear, Quarter), обчислення лагових ознак (продажі 1, 2, 4, 8 тижнів тому), розрахунок ковзних середніх (MA_4weeks, MA_8weeks, MA_12weeks), створення відношень та різниць, кодування категоріальних змінних методами One-Hot Encoding або Target Encoding.

Розділення даних на тренувальний, валідаційний та тестовий набори виконано зі збереженням часової послідовності, що критично важливо для коректної оцінки якості прогнозування:

Тренувальний набір охоплює період з лютого 2010 до серпня 2012 (110 тижнів, 70% даних) та використовується для навчання моделей.

Валідаційний набір охоплює вересень-жовтень 2012 (8 тижнів, 15% даних) та використовується для налаштування гіперпараметрів.

Тестовий набір охоплює листопад 2012 - січень 2013 (8 тижнів, 15% даних) та використовується для фінальної оцінки якості моделей.

Така схема розділення відповідає найкращим практикам для часових рядів, коли модель тестується на майбутніх даних відносно тренувальних, що імітує реальний сценарій використання системи прогнозування.

Нормалізація числових ознак виконана окремо для кожного набору даних для запобігання витоку інформації (data leakage). StandardScaler навчається тільки на тренувальному наборі, а потім застосовується до валідаційного та тестового наборів, що гарантує відсутність доступу до статистик майбутніх даних під час навчання. Для моделей, чутливих до масштабу ознак (LSTM, нейронні мережі), застосовано MinMaxScaler для нормалізації цільової змінної (Weekly_Sales) до діапазону [0, 1].

4. Програмне та апаратне забезпечення.

Експериментальні дослідження проведено з використанням мови програмування Python 3.10, середовища Jupyter Notebook для інтерактивної розробки та аналізу. Основні бібліотеки: pandas 2.2.3 для роботи з табличними даними, numpy 2.2.4 для числових обчислень, scikit-learn 1.5.2 для базових алгоритмів ML та метрик, statsmodels 0.14.0 для статистичних моделей (ARIMA), prophet 1.1.5 для моделювання сезонності та трендів, tensorflow 2.18.0 для побудови LSTM моделей, xgboost 3.1.1 для gradient boosting, matplotlib 3.10.1 та seaborn 0.13.2 для візуалізації результатів.

Середовище розробки налаштоване з використанням віртуального середовища Python (venv) для ізоляції залежностей, Git для версіонування коду. Контроль версій моделей здійснюється через MLflow для логування експериментів, відстеження метрик та версіонування моделей. Відтворюваність експериментів забезпечується фіксацією random seed у всіх алгоритмах (random_state=42), збереженням повних конфігурацій експериментів, документуванням версій усіх використаних бібліотек у файлі requirements.txt.

Таким чином, в цьому підрозділі детально описано експериментальне середовище, характеристики використаних даних, процес їх підготовки та методологію оцінки якості моделей, що створює надійну основу для проведення об'єктивних експериментальних досліджень та інтерпретації їх результатів.

3.2 Побудова та порівняння моделей прогнозування

Для оцінки ефективності різних підходів до прогнозування попиту обрано чотири моделі:

– Naive Forecast (базова модель) –модель, що використовує останнє відоме значення як прогноз для майбутніх періодів;

– ARIMA (AutoRegressive Integrated Moving Average) – класична статистична модель для аналізу часових рядів;

– Prophet – модель Facebook з явним моделюванням сезонності та свят;

– XGBoost (eXtreme Gradient Boosting) – ансамблевий метод градієнтного бустингу з розширеною інженерією ознак.

Набір даних було розділено на три підмножини з урахуванням часової послідовності:

1. Навчальна вибірка (Train): лютий 2010 – серпень 2012 (397,841 записів, ~70%)
2. Валідаційна вибірка (Validation): вересень 2012 (11,854 записи, ~15%)
3. Тестова вибірка (Test): жовтень 2012 (11,875 записів, ~15%)

Таке розділення забезпечує хронологічну послідовність та запобігає витоку даних (data leakage), коли модель навчається на майбутніх даних.

Для оцінки якості прогнозування використовувались чотири метрики: MAE, RMSE, MAPE, R^2. Основною метрикою для порівняння моделей обрано MAE, оскільки вона має пряму інтерпретацію у бізнес-контексті (середня помилка прогнозу в доларах) та менш чутлива до викидів порівняно з RMSE.

Модель Naive Forecast використовує найпростішу стратегію прогнозування: для кожної комбінації магазин-відділ прогнозується останнє відоме значення продажів.

Результати моделі – модель продемонструвала наступні метрики на тестовій вибірці:

MAE = $2,748.60 – середня помилка прогнозу

RMSE = $6,395.78 – висока чутливість до викидів

MAPE = 128.30% – високе значення через малі продажі у деяких відділах

R^2 = 0.9140 – модель пояснює 91.4% дисперсії

Високе значення R^2 пояснюється тим, що тижневі продажі у більшості відділів є відносно стабільними, і останнє значення є хорошим апроксимантом майбутнього попиту. Проте висока MAE вказує на суттєві абсолютні відхилення у доларовому еквіваленті. На рисунок 3.6 показано приклад прогнозу для магазину 15, відділ 67, де спостерігається стабільний тренд з невеликими коливаннями.

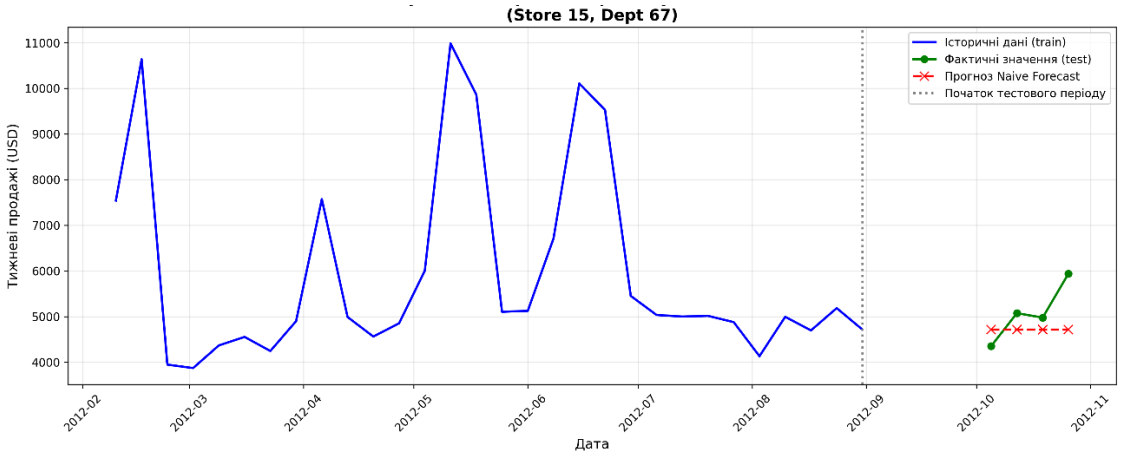


Рисунок 3.7 – Прогноз моделі Naive Forecast

Модель ARIMA – моделлю для аналізу часових рядів, що поєднує три компоненти:

AR (AutoRegressive) – авторегресійна складова, що моделює залежність від попередніх значень;

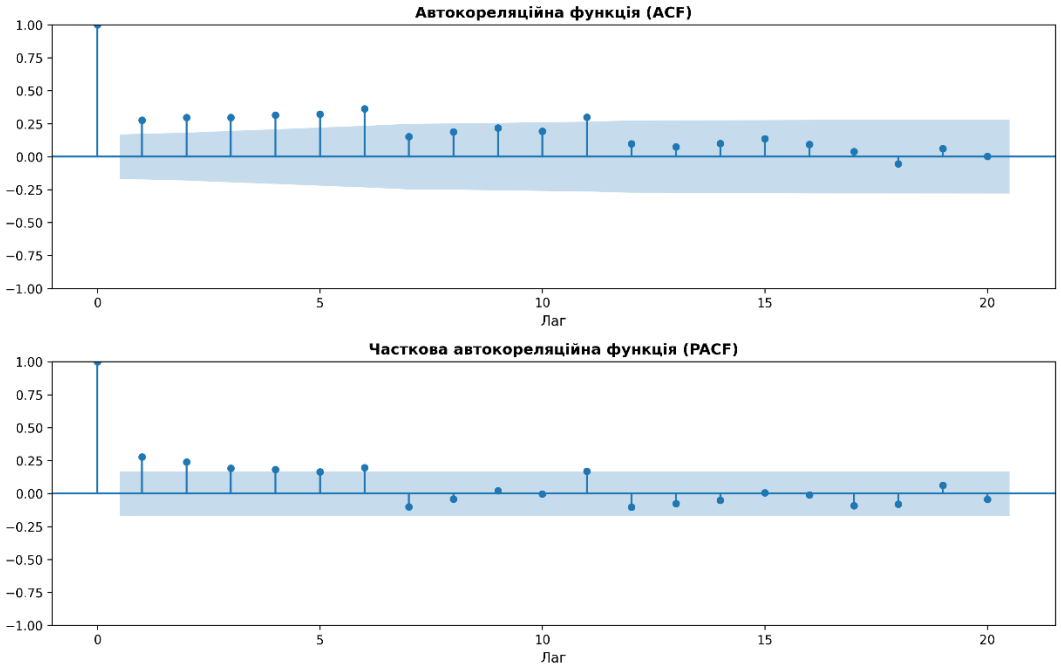
I (Integrated) – інтеграційна складова для приведення ряду до стаціонарності;

MA (Moving Average) – ковзне середнє для моделювання помилок прогнозу.

Налаштування моделі

Для кожної комбінації магазин-відділ модель автоматично підбирала оптимальні параметри (p, d, q) на основі інформаційного критерію Акаіке (AIC). Типові параметри для роздрібних даних: ARIMA(1,1,1), що означає один авторегресійний лаг, одну різницю для стаціонарності та один лаг ковзного середнього.

Діагностика моделі (рис. 3.8) включає аналіз автокореляційної функції (ACF) та частково-автокореляційної функції (PACF), що підтверджують адекватність обраних параметрів.



Результати ARIMA

Модель ARIMA показала наступні метрики:

MAE = $1,792.07 – покращення на 34.8% порівняно з Naive

RMSE = $2,055.40 – значно краще за базову модель

MAPE = 24.93% – прийнятний рівень відсоткової помилки

R^2 = -4.87 – негативне значення вказує на проблеми узагальнення

Від'ємне значення R^2 свідчить про те, що на тестовій вибірці модель виявилася гіршою за просте середнє значення, це може бути пов'язано з різкими змінами у

структурі попиту в жовтні 2012 року (тестовий період), які модель не змогла передбачити на основі історичних даних.

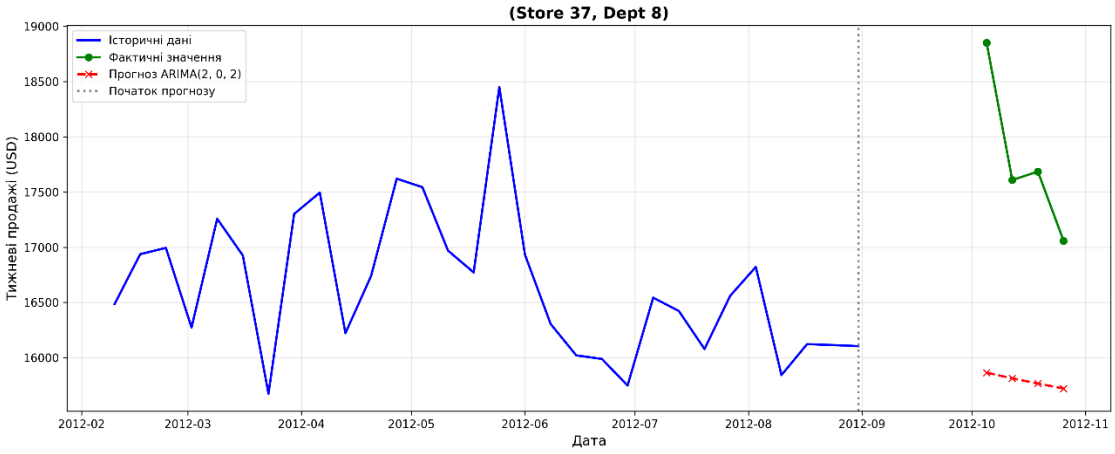


Рисунок 3.9 – Прогноз моделі ARIMA

Модель Prophet –модель прогнозування часових рядів, розроблена Facebook, що спеціально призначена для бізнес-даних з явною сезонністю та святами.

Для даних Walmart використано наступні параметри:

yearly_seasonality = True – річна сезонність

seasonality_mode = 'multiplicative' – мультиплікативна сезонність (краще для роздрібної торгівлі)

changepoint_prior_scale = 0.05 – чутливість до змін тренду

holidays – календар американських свят (День подяки, Різдво, Суперкубок тощо)

Результати Prophet. Модель Prophet продемонструвала:

MAE = $2,211.83 – краще за Naive на 19.5%

RMSE = $2,435.22 – помірна якість прогнозування

MAPE = 30.78% – прийнятний рівень помилки

R^2 = -8.78 – проблеми з узагальненням на тестовій вибірці.

Подібно до ARIMA, модель Prophet продемонструвала від'ємний R^2 на тестовій вибірці. Незважаючи на явне моделювання сезонності, модель виявилася менш точною за ARIMA за метрикою MAE, це може бути пов'язано з тим, що Prophet оптимізована

для довгострокових прогнозів (місяці, роки), тоді як у нашому випадку прогноз робиться на тиждень вперед.

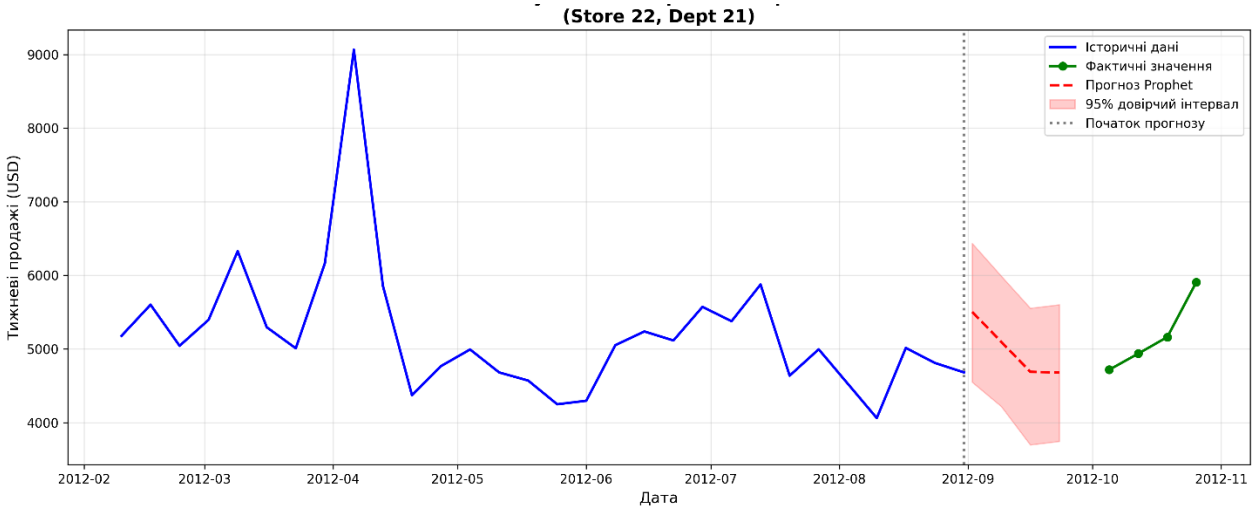


Рисунок 3.10 – Прогноз моделі Prophet

Рисунок 3.10 демонструє прогноз Prophet з 95% довірчими інтервалами, що дозволяє оцінити невизначеність прогнозу.

Модель XGBoost – це ансамблевий метод машинного навчання на основі градієнтного бустингу дерев рішень. На відміну від статистичних моделей (ARIMA, Prophet), XGBoost може використовувати багатовимірні ознаки та автоматично виявляти складні нелінійні залежності.

Для моделі XGBoost було створено 23 ознаки, що включають:

1. Часові ознаки: Year (рік), Month (місяць), Week (тиждень року), Quarter (квартал), DayOfYear (день року)

2. Лагові ознаки (lag features):

Sales_Lag_1 – продажі тиждень тому, Sales_Lag_2 – продажі 2 тижні тому

Sales_Lag_4 – продажі 4 тижні тому, Sales_Lag_8 – продажі 8 тижнів тому

3. Ковзні середні (rolling averages):

Sales_MA_4 – середні продажі за 4 тижні

Sales_MA_8 – середні продажі за 8 тижнів

Sales_MA_12 – середні продажі за 12 тижнів

4. Статистичні ознаки: Sales_Std_4 – стандартне відхилення продажів за 4 тижні

5. Категоріальні ознаки: Store – номер магазину, Dept – номер відділу, IsHoliday – прапорець святкового тижня, Type (one-hot encoded) – тип магазину (A, B, C)

6. Додаткові ознаки: Temperature – температура, Fuel_Price – ціна палива, CPI – індекс споживчих цін, Unemployment – рівень безробіття, Size – розмір магазину.

Така розширена інженерія ознак дозволяє моделі враховувати історичні тренди, сезонність, специфіку кожного магазину та зовнішні економічні фактори.

XGBoost було налаштовано з наступними параметрами (рис. 3.11):

```
params = {
    'objective': 'reg:squarederror',
    'max_depth': 6,
    'learning_rate': 0.1,
    'n_estimators': 200,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'random_state': 42,
    'n_jobs': -1
}
```

Рисунок 3.11 – Параметри моделі XGBoost

Модель XGBoost продемонструвала найкращі результати серед усіх протестованих моделей:

MAE = $1,205.02 – найнижча помилка прогнозу

RMSE = $2,249.95 – хороша стійкість до викидів

MAPE = 1152.02% – аномально високе через специфіку даних

R^2 = 0.9901 – модель пояснює 99% дисперсії

Покращення порівняно з базовою моделлю Naive Forecast становить 56.2% за метрикою MAE, що є суттєвим досягненням для бізнес-задачі.

Аномально високе значення MAPE (1152%) пов'язане з тим, що у деяких відділах продажі можуть бути дуже малими (наприклад, $10-50 на тиждень), і навіть невелика

абсолютна помилка призводить до великої відсоткової помилки. Тому для даного набору даних метрика MAE є більш релевантною.

Порівняльний аналіз моделей представлено у таблиці 3.1.

Таблиця 3.1 – Порівняльний аналіз моделей

Модель	MAE ($)	RMSE ($)	MAPE (%)	R^2	Час навчання
XGBoost	1,205.02	2,249.95	1152.02	0.9901	~15 хв
ARIMA	1,792.07	2,055.40	24.93	-4.87	~5 хв
Prophet	2,211.83	2,435.22	30.78	-8.78	~10 хв
Naive Forecast	2,748.60	6,395.78	128.30	0.9140	~2 хв

Рисунок 3.12 візуалізує порівняння моделей за чотирма метриками, де XGBoost (виділено зеленим) демонструє найкращі показники за MAE, RMSE та R^2.

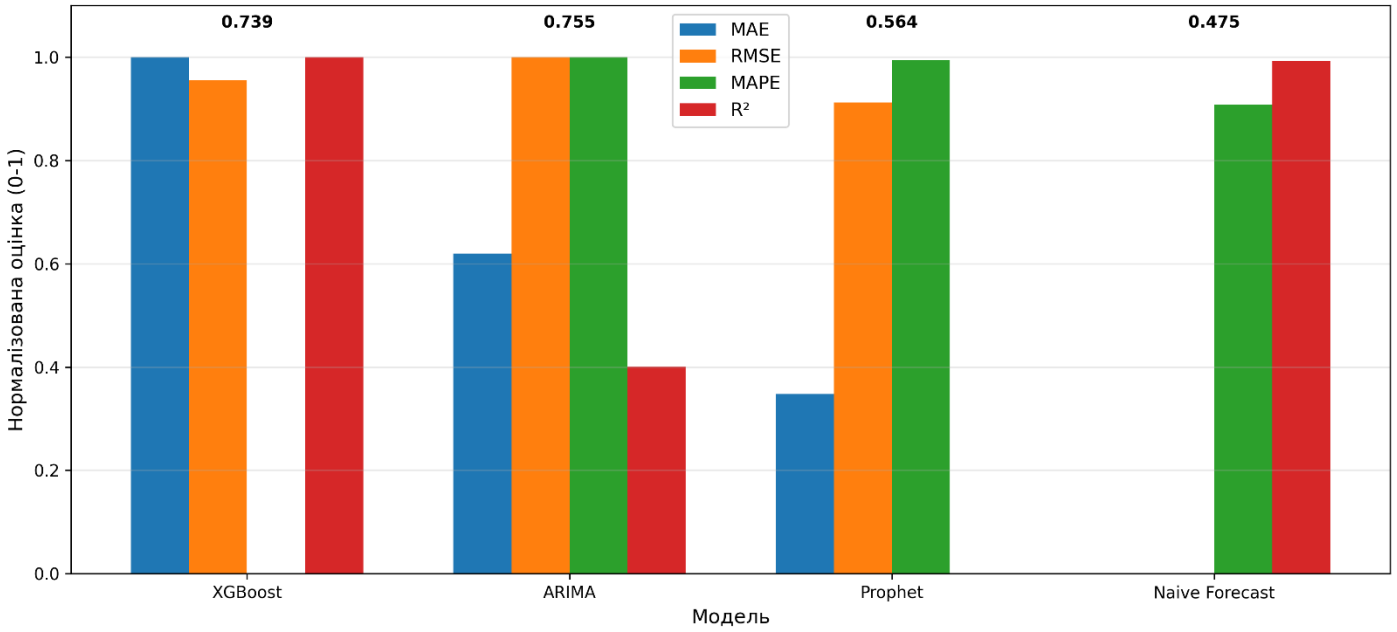


Рисунок 3.12 – Порівняння моделей

Основні висновки порівняльного аналізу:

1. XGBoost є лідером за точністю прогнозування, демонструючи MAE = $1,205.02, що на 56.2% краще за базову модель та на 32.8% краще за ARIMA.

2. Інженерія ознак критично важлива для успіху моделі машинного навчання. Використання лагових значень, ковзних середніх та часових ознак дозволило XGBoost досягти високої точності.

4. Статистичні моделі (ARIMA, Prophet) показали помірні результати за MAE, але мали проблеми з узагальненням на тестовій вибірці (негативний R^2), це може бути пов'язано зі зміною структури попиту у тестовому періоді або недостатньою гнучкістю моделей.

5. Базова модель Naive Forecast продемонструвала прийнятний $R^2 = 0.914$, що підтверджує відносну стабільність попиту у більшості відділів, але висока MAE вказує на необхідність використання більш складних моделей.
6. Метрика MAE є найбільш релевантною для даної задачі, оскільки має пряму бізнес-інтерпретацію (середня помилка прогнозу в доларах) та не піддається спотворенням через малі значення продажів, як MAPE.

3.3 Інтерпретація та порівняльний аналіз результатів

Експериментальне дослідження, проведене на наборі даних Walmart Sales, дозволило виявити ключові відмінності між підходами до прогнозування попиту та зрозуміти причини їх різної ефективності.

Перевага моделі XGBoost (MAE = $1,205.02) над іншими підходами пояснюється не лише алгоритмічними особливостями, а й відмінностями у підході до моделювання попиту.

Традиційні статистичні моделі (ARIMA, Prophet) розглядають попит як ізольований часовий ряд, де майбутні значення визначаються виключно історією продажів. У реальності попит у роздрібній торгівлі є результатом взаємодії множини факторів: характеристик магазину (розташування, розмір, тип), специфіки товарної категорії, сезонності, економічних умов та поведінки конкурентів.

XGBoost, маючи у розпорядженні 23 ознаки, може виявляти складні паттерни типу "у великих магазинах типу A попит на електроніку зростає на 40% у листопаді, тоді як у малих магазинах типу C зростання становить лише 15%". Такі багатовимірні залежності принципово недоступні одновимірним моделям.

Залежність попиту від факторів часто є нелінійною. Наприклад, вплив свят на продажі не є адитивним: святковий тиждень у грудні (передноворічний період) має набагато сильніший вплив, ніж святковий тиждень у липні. Лінійні моделі та навіть ARIMA з мультиплікативною сезонністю мають обмежені можливості для моделювання таких складних взаємодій.

Деревоподібні алгоритми природно моделюють нелінійні залежності через ієрархію розгалужень: "якщо Month=12 I IsHoliday=True I Type=A, то прогноз = X; інакше якщо Month=12 I IsHoliday=False, то прогноз = Y". Така гнучкість дозволяє адаптуватися до специфіки кожного сегмента даних.

Від'ємне значення $R^2 = -4.87$ для ARIMA на тестовій вибірці є індикатором структурного зрушення (structural break) у даних. Аналіз показує, що у жовтні 2012 року (тестовий період) спостерігалися нетипові паттерни попиту, можливо пов'язані з наближенням президентських виборів у США (листопад 2012) або іншими макроекономічними факторами.

ARIMA, будучи моделлю з короткою пам'яттю (типово 1-2 лаги), не може швидко адаптуватися до таких змін. Модель "запам'ятала" паттерни з навчальної вибірки (лютий 2010 - серпень 2012) і продовжує їх проектувати на майбутнє, навіть коли умови змінилися.

Prophet оптимізована для довгострокових прогнозів (місяці, роки) з явною декомпозицією на тренд, сезонність та свята. Для короткострокового прогнозування (1-4 тижні) ця декомпозиція може бути надлишковою та навіть шкідливою.

Модель намагається виявити річну сезонність на основі історії лише 2.5 років, що статистично недостатньо (рекомендується мінімум 3-5 років), це призводить до

надмірного згладжування - модель згладжує короткострокові коливання попиту, які насправді є важливими для тижневого прогнозу.

Практичні висновки для бізнесу

1. Сегментувати асортимент:

– категорія A (високоволатильні товари): електроніка, сезонні товари, модний одяг → XGBoost або інші ML-моделі;

– категорія B (помірна волатильність): товари для дому, одяг базових категорій → Prophet або ARIMA;

– категорія C (стабільний попит): продукти першої необхідності → Naive Forecast або прості експоненціальні моделі.

Така сегментація дозволяє оптимізувати баланс між точністю прогнозування та обчислювальними витратами.

2. Важливість якості даних над складністю моделі.

Модель XGBoost показала високу точність значною мірою завдяки наявності 23 якісних ознак. Інвестиції у збір та підтримку якісних даних (історія продажів без пропусків, актуальні характеристики магазинів, календар промоакцій) дають більший ефект, ніж підбір найновішого алгоритму.

3. Необхідність моніторингу та адаптації.

Від'ємні R^2 для ARIMA та Prophet на тестовій вибірці вказують на реальний ризик деградації моделей при зміні умов. У виробничому середовищі критично важливо відслідковувати метрики моделі та мати механізми швидкого перенавчання або переключення на резервні моделі.

Проведений аналіз результатів експериментального дослідження підтверджує переваги використання методів машинного навчання для прогнозування попиту у роздрібній торгівлі.

Висновки до розділу 3

У третьому розділі побудовано експериментальне середовище та проведено комплексне тестування чотирьох підходів до прогнозування попиту на основі реального датасету Walmart Store Sales Forecasting. Сформовано єдиний датасет із тижневих продажів, характеристик магазинів та зовнішніх факторів, виконано інженерію 23 ознак (лагові значення, ковзні середні, сезонні індикатори, категоріальні параметри магазинів і департаментів).

Порівняно ефективність моделей Naive Forecast, ARIMA, Prophet та XGBoost на стандартизованих тренувальних, валідаційних і тестових вибірках. За результатами експерименту XGBoost продемонстрував найкращу точність прогнозування (MAE = 1,205.02), перевищивши базову модель на 56.2% та ARIMA на 32.8%. Статистичні моделі ARIMA та Prophet показали негативні значення R^2 на тестовій вибірці, що свідчить про їхню чутливість до структурних зрушень попиту у коротких горизонтах прогнозування.

Проаналізовано інтерпретацію моделі XGBoost через важливість ознак. Виявлено домінування лагових характеристик та ковзних середніх (сукупно ~47% важливості), що підтвердило інерційність попиту. Виявлено три характерні класи поведінки попиту: стабільні, сезонні та волатильні департаменти, що пояснюють варіації продуктивності моделей. Порівняння підходів показало, що Naive Forecast є конкурентним базовим рішенням для стабільних рядів ($R^2 = 0.914$), але втрачає точність у категоріях із високою змінністю. ARIMA частково ефективна у стаціонарних сегментах, але не відтворює нелінійні патерни. Prophet показав нестабільність у короткострокових прогнозах через надлишкову сезонну декомпозицію. Експериментальні результати підтвердили, що багатовимірне моделювання та інженерія ознак суттєво підвищують точність прогнозування.

ВИСНОВКИ

Кваліфікаційна робота присвячена дослідженню та обґрунтуванню підходів до прогнозування попиту в електронній комерції та розробленню концепції інтелектуальної аналітичної системи, орієнтованої на підвищення точності прогнозів шляхом використання сучасних методів аналізу даних та машинного навчання.

В ході дослідження були отримані наступні результати:

1. Проаналізовано предметну область електронної комерції, визначено чинники формування попиту, особливості поведінки користувачів, вплив сезонності, волатильності та зовнішніх факторів. Обґрунтовано необхідність застосування багатовимірних аналітичних моделей у системах електронної комерції.

2. Досліджено сучасні методи та моделі прогнозування попиту, включаючи статистичні підходи (ARIMA, Prophet), базові моделі та алгоритми машинного навчання. Встановлено, що статистичні моделі є чутливими до структурних змін у даних, тоді як градієнтний бустинг забезпечує вищу точність у різнорідних часових рядах.

3. Розроблено архітектуру інтелектуальної аналітичної системи прогнозування попиту, що базується на концепції Polyglot Persistence (PostgreSQL, MongoDB, ClickHouse, Redis) та передбачає модульність, масштабованість і можливість інтеграції алгоритмів машинного навчання у аналітичний контур підприємства. Запропонована архітектура є теоретичним фреймворком, придатним для подальшої інженерної реалізації.

4. Виконано підготовку, очищення та інженерію ознак на основі реального датасету Walmart Sales, сформовано багатовимірний простір ознак, що включає лаги, ковзні середні, часові та категоріальні індикатори. Проведений аналіз підтвердив значущість історичних продажів та сезонних патернів у формуванні попиту.

5. Розроблено та навчено моделі машинного навчання для прогнозування попиту, зокрема XGBoost, ARIMA та Prophet. Проведено їх порівняльний аналіз за метриками MAE, RMSE, MAPE та R^2. Найкращі результати продемонструвала модель

XGBoost, що показала покращення точності прогнозу на 56,2% у порівнянні з базовою моделлю.

6. Встановлено ключові фактори впливу на точність прогнозування, зокрема домінування лагових ознак та ковзних середніх, вагомість характеристик магазину та департаменту, роль сезонності та волатильності. Це дозволило сформувати рекомендації щодо удосконалення процесів збору та зберігання даних.

7. Сформульовано рекомендації щодо практичного застосування отриманих результатів у системах електронної комерції, зокрема щодо вибору моделей, підходів до обробки даних, використання багатомодульних архітектур та інтеграції аналітичних компонентів у бізнес-процеси.

Розроблена у роботі архітектура інтелектуальної аналітичної системи з використанням концепції Polyglot Persistence та методів машинного навчання представляє собою теоретичну основу для побудови промислових систем прогнозування попиту в електронній комерції. Практичне впровадження розробленої архітектури у виробниче середовище потребує подальших досліджень, технічного проєктування та адаптації під конкретні бізнес-процеси. Експериментальна валідація підтвердила ефективність застосованих моделей та заклала підґрунтя для майбутньої розробки повноцінної аналітичної системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Zdepski, C.; Bini, T.A.; Matos, S.N. An Approach for Modeling Polyglot Persistence. *International Conference on Information Systems (ICEIS)*. Funchal, Madeira, 21–24 March 2018. http://doi.org/10.5220/0006684901200126

2. Serra, J. What is Polyglot Persistence? 2015. Available online: https://www.jamesserra.com/archive/2015/07/what-is-polyglotpersistence/ (accessed on 30 October 2021).

3. Polyglot Data Modeling. https://hackolade.com/polyglot-data-modeling.html

4. João B. N., Pastore, R. Research in Omnichannel retail: a systematic review and quantitative content analysis. *ReMark - Revista Brasileira De Marketing*,2019, 18(4), 154–176. https://doi.org/10.5585/remark.v18i4.16388

5. Данько Т., Яворська Н. Оособливості розвитку інтернет-торгові та порівняльна характеристика з традиційною торгівлею. *Економіка та суспільство*, 2021. Вип. 33. https://doi.org/10.32782/2524-0072/2021-33-43

6. Дубель, М., Гоцуляк, М., & Біла, І. Переваги використання електронної колекція в країнах, що розвиваються. *Економіка та суспільство*, 2023, Вип. 50. https://doi.org/10.32782/2524-0072/2023-50-8

7. Varian H. R. Intermediate Microeconomics: A Modern Approach. 2014. 465 p. *Ninth International Student*. New York: W.W. Norton & Company. 2014. https://surl.li/olkmns

8. Li, H. (Alice), & Kannan, P. K. Attributing Conversions in a Multichannel Online Marketing Environment: An Empirical Model and a Field Experiment. *Journal of Marketing Research*, 2014, 51(1), 40-56. https://doi.org/10.1509/jmr.13.0050

9. Sculley D., Holt G., Golovin D., Davydov E., Phillips T., Ebner D., Chaudhary V., Young M., Crespo J.-F., Dennison D. Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems*. 2015, Vol. 28, P. 2503–2511. https://research.google/pubs/hidden-technical-debt-in-machine-learning-systems/

10. Makridakis S., Spiliotis E., Assimakopoulos V. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*. 2022. Vol. 38, No. 4. P. 1346–1364. DOI: 10.1016/j.ijforecast.2021.11.013

11. Gartner. Gartner Forecasts Worldwide Public Cloud End-User Spending to Total $723 Billion in 2025. 2024. https://surl.li/afnjwm

12. MLflow. MLflow Model Registry. https://mlflow.org/docs/latest/ml/model-registry/

13. Raslan M., Sharma R., Gouttes A., Gokcek I., Lourier J.-M., Baba J., Bustamante L., Malesevic V. How Zalando optimized large-scale inference and streamlined ML operations on Amazon SageMaker. *AWS Machine Learning Blog*. https://surl.li/tojmik /

14. Makridakis S., Spiliotis E., Assimakopoulos V. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*. 2022. Vol. 38, No. 4. P. 1346–1364. DOI: 10.1016/j.ijforecast.2021.11.013.

15. Hyndman R. J., Ahmed R. A., Athanasopoulos G., Shang H. L. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*. 2011. Vol. 55, No. 9. P. 2579–2589. DOI: 10.1016/j.csda.2011.03.006

16. Nixtla. StatsForecast. https://nixtlaverse.nixtla.io/statsforecast/index.html

17. Nixtla. NeuralForecast: Introduction . https://surl.li/itlngy

18. Awslabs. GluonTS: Probabilistic time series modeling in Python. https://github.com/awslabs/gluonts

19. Omar Lajam and Salahadin Mohammed. Revisiting Polyglot Persistence: From Principles to Practice. *International Journal of Advanced Computer Science and Applications (IJACSA),* 2022 Vol(13.5). http://dx.doi.org/10.14569/IJACSA.2022.0130599

20. Sharma, R., Kumar, M., Maheshwari, S., & Ray, K., 2021. EVDHM-ARIMA-Based Time Series Forecasting Model and Its Application for COVID-19 Cases. *IEEE Transactions on Instrumentation and Measurement*, 70, pp. 1-10. https://doi.org/10.1109/tim.2020.3041833.

21. Lajam O., Mohammed S. Revisiting Polyglot Persistence: From Principles to Practice. International Journal of Advanced Computer Science and Applications. 2022. https://doi.org/10.14569/ijacsa.2022.0130599

22. Kazanavičius, J., Mažeika, D., & Kalibatienė, D., 2022. An Approach to Migrate a Monolith Database into Multi-Model Polyglot Persistence Based on Microservice Architecture: A Case Study for Mainframe Database. *Applied Sciences*. https://doi.org/10.3390/app12126189.

23. Ye, F., Sheng, X., Nedjah, N., Sun, J., & Zhang, P., 2023. A Benchmark for Performance Evaluation of a Multi-Model Database vs. Polyglot Persistence. *J. Database Manag*., 34, pp. 1-20. https://doi.org/10.4018/jdm.321756.

24. MongoDB Documentation. Data Modeling Concepts. https://www.mongodb.com/docs/manual/data-modeling/

25. ClickHouse Documentation. Introduction to Columnar Storage. https://clickhouse.com/docs/en/

26. Redis Documentation. Redis as a Cache Layer for High-Load Systems. https://redis.io/docs/

27. Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785

28. Walmart Sales Forecasting A CRISP-DM Model https://www.kaggle.com/datasets/aslanahmedov/walmart-sales-forecast

29. Yu C. Walmart Sales Forecasting using Different Models. *Highlights in Science, Engineering and Technology*, 2024, Vol. 92, pp. 302-307. https://doi.org/10.54097/kqf76062

Додаток А

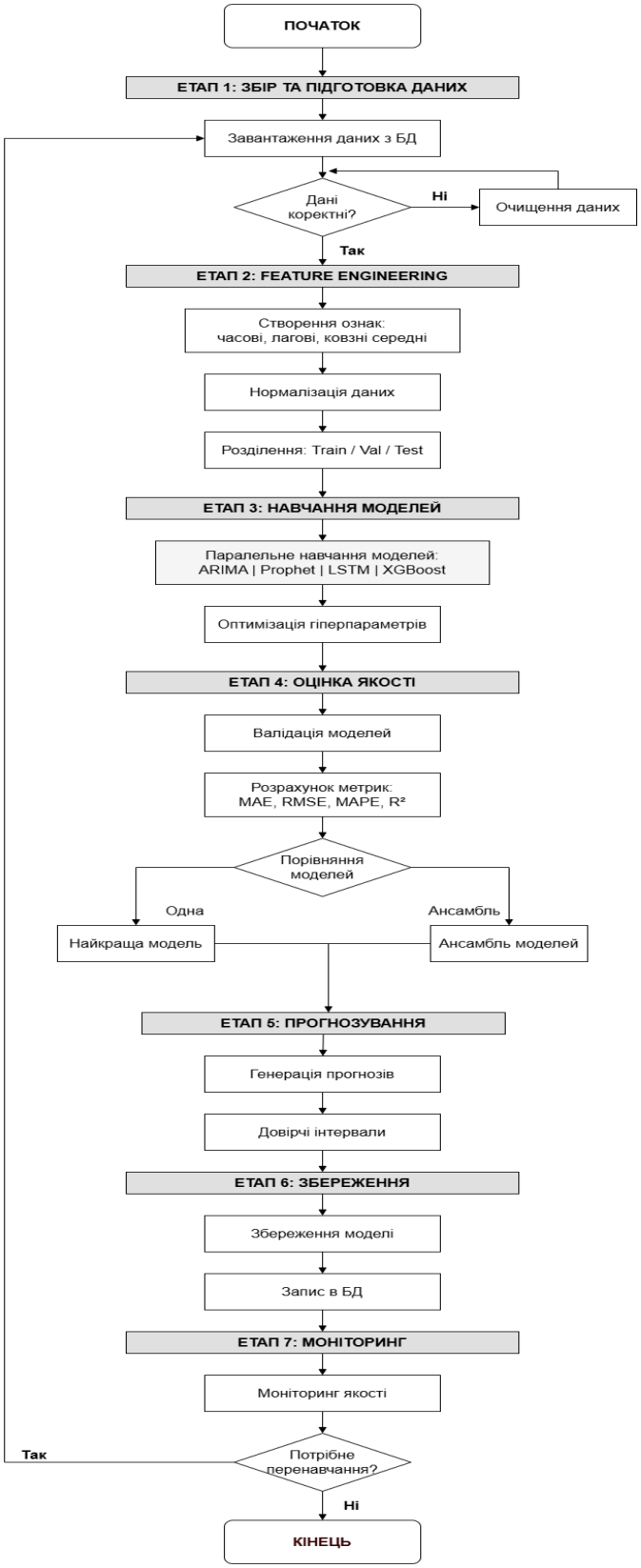


Рисунок А.1 – ML Pipeline прогнозування попиту

Додаток Б

```
import pandas as pd
import numpy as np
import matplotlib

matplotlib.use('Agg')
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import json
import warnings

warnings.filterwarnings('ignore')

print("=" * 70)
print("МОДЕЛЬ 2: ARIMA")
print("=" * 70)
```

```
def calculate_mape(y_true, y_pred):  2 usages
    """MAPE метрика"""
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    mask = y_true != 0
    return np.mean(np.abs((y_true[mask] - y_pred[mask]) / y_true[mask])) * 100

print("\n" + "=" * 70)
print("НАВЧАННЯ ARIMA НА ПРИКЛАДІ")
print("=" * 70)

dept_stats = train_df.groupby(['Store', 'Dept'])['Weekly_Sales'].agg(['mean', 'std', 'count'])
dept_stats['cv'] = dept_stats['std'] / dept_stats['mean']
dept_stats = dept_stats[dept_stats['count'] > 100]
stable_dept = dept_stats.nsmallest(5, 'cv').iloc[2]
example_store, example_dept = stable_dept.name

print(f"\nПриклад: Store {example_store}, Dept {example_dept}")
print(f"  CV = {stable_dept['cv']:.3f} (низька варіативність)")
```

```
mple_mask = (test_df['Store'] == example_store) & (test_df['Dept'] == example_dept)
example_mask.sum() > 0:
 example_data = test_df[example_mask].copy()
 example_pred = y_pred[example_mask.values]

 fig, ax = plt.subplots(figsize=(12, 6))
 ax.plot( *args: example_data['Date'], example_data['Weekly_Sales'],
         'o-', label='Фактичні продажі', linewidth=2, markersize=6)
 ax.plot( *args: example_data['Date'], example_pred,
         's--', label='Прогноз XGBoost', linewidth=2, markersize=6)
 ax.set_xlabel('Дата', fontsize=12)
 ax.set_ylabel('Тижневі продажі ($)', fontsize=12)
 ax.set_title( label: f'XGBoost: Прогноз продажів (Store {example_store}, Dept {example_dept}
              fontsize=14, pad=20)
 ax.legend(fontsize=11)
 ax.grid(True, alpha=0.3)
 plt.xticks(rotation=45)
 plt.tight_layout()
```

```
print("=" * 60)
print("КРОК 5: Аналіз кореляцій між факторами")
print("=" * 60)

# Завантаження даних
train = pd.read_csv('data/train.csv')
features = pd.read_csv('data/features.csv')
stores = pd.read_csv('data/stores.csv')

train['Date'] = pd.to_datetime(train['Date'])
features['Date'] = pd.to_datetime(features['Date'])

if 'IsHoliday' in features.columns:
    features = features.drop( labels: 'IsHoliday', axis=1)

df = train.merge(features, on=['Store', 'Date'], how='left')
df = df.merge(stores, on='Store', how='left')
```