

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ЦИНАЙКО Василь Петрович

Програмний модуль аналізу та прогнозування продажів аптечної мережі / Software Module for Pharmaceutical Sales Analysis and Forecasting

спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН–43
В. П. Цинайко

Науковий керівник:
к.т.н., доцент, П. Є. Биковий

Кваліфікаційну роботу
допущено до захисту:
" ____ " _____ 20 ____ р.

Завідувач кафедри
_____ Н. В. Дзюбановська

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 «Комп'ютерні науки»
освітньо-професійна програма – «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»
В. о. завідувача кафедри ІОСУ
_____ Н. В. Дзюбановська
«_____» _____ 20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЦИНАЙКО Василю Петровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Програмний модуль аналізу та прогнозування продажів аптечної мережі /
Software Module for Pharmaceutical Sales Analysis and Forecasting

керівник роботи _____ **к.т.н., доцент, П. Є. Биковий,**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “01” грудня 2025 року № 894

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи завдання на кваліфікаційну роботу
студента, наукові статті, технічна література

4. Основні питання, які потрібно розробити:

- Проаналізувати особливості предметної області та дослідити існуючі методи аналізу й прогнозування часових рядів у сфері роздрібної торгівлі медикаментами.
- Сформулювати конвеєр попередньої обробки та інженерії ознак (Feature Engineering) для очищення транзакційних даних від аномалій та врахування екзогенних бізнес-факторів.
- Обґрунтувати вибір та здійснити математичне налаштування предиктивного алгоритму побудови прогнозів (моделі Prophet).
- Спроектувати об'єктно-орієнтовану архітектуру та реалізувати вебзастосунок аналітичного дашборду на базі сучасного технологічного стека (Python, Streamlit, Plotly).
- Провести функціональне тестування розробленого програмного комплексу та оцінити точність прогнозової моделі за допомогою метрик Percentage Error (Зокрема MAPE).

5. Перелік графічного матеріалу у роботі:

- Візуалізація архітектури модуля.
- Візуалізація внутрішнього і зовнішнього впливу на багатші дані.
- Структурна схема програмної системи.
- UML-діаграма класів програмної системи прогнозування.
- UML діаграма станів графічного інтерфейсу дашборду.
- Візуалізація результатів прогнозування.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент

_____ В. П. Цинайко
(підпис) (прізвище та ініціали)

Керівник роботи

_____ к.т.н., доцент П. Є. Биковий
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль аналізу та прогнозування продажів аптечної мережі» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом у 67 сторінок і містить 6 ілюстрацій, 2 додатки та 25 використаних джерел.

Об'єктом дослідження є процеси операційного планування та предиктивного аналізу часових рядів обсягів реалізації товарів у роздрібній торгівлі.

Метою роботи є розробка інтерактивного програмного модуля (MVP-дашборду) для аналізу та предиктивного моделювання продажів фармацевтичної продукції на основі алгоритмів машинного навчання для оптимізації товарних запасів аптечної мережі.

Методами дослідження та розроблення обрано: методи об'єктно-орієнтованого проектування (UML-моделювання), предиктивний аналіз часових рядів за допомогою алгоритму Prophet, а також емпіричні методи валідації та оцінки якості моделей машинного навчання (MAPE, MSE).

Внаслідок виконання роботи спроектовано та реалізовано веб-сторінку предиктивної аналітики роздрібного попиту на базі алгоритмів машинного навчання. Розроблений програмний модуль автоматизує розрахунок майбутньої потреби в медикаментах, що дозволяє суттєво знизити вплив людського фактору, оптимізувати обсяги поставок ліків та мінімізувати ризики «заморожування» обігових коштів підприємства в надлишкових товарних запасах.

Результати дослідження мають практичну цінність для аналітичних підрозділів і департаментів постачання роздрібних фармацевтичних мереж, а також можуть застосовуватися як інструмент підтримки прийняття рішень у системах операційного планування.

Ключові слова: ЧАСОВІ РЯДИ, ПРОГНОЗНЕ МОДЕЛЮВАННЯ, МОДЕЛЬ PROPHET, PYTHON, STREAMLIT, ІНТЕРАКТИВНИЙ ДАШБОРД, MAPE METRIC, ОПЕРАТИВНЕ ПЛАНУВАННЯ, ОПТИМІЗАЦІЯ ЗАПАСІВ.

ANNOTATION

The qualification work on the topic “Software Module for Pharmaceutical Sales Analysis and Forecasting” for obtaining a Bachelor’s degree in specialty 122 “Computer Science” of the “Computer Science” educational program consists of 67 pages and contains 6 illustrations, 2 appendices, and 25 references.

The subject of this study is the processes of operational planning and predictive analysis of time series data on retail sales volumes.

The purpose of the work is to develop an interactive software module (MVP dashboard) for the analysis and predictive modeling of pharmaceutical sales based on machine learning algorithms to optimize the inventory of a pharmacy chain.

Research and development methods include: object-oriented design methods (UML modeling), time-series predictive analysis using the Prophet algorithm, as well as empirical methods for validation and quality assessment of machine learning models (MAPE, MSE).

As a result of the work, a predictive analytics web application for retail demand based on machine learning algorithms was designed and implemented. The developed software module automates the calculation of future pharmaceutical demands, significantly reducing the impact of the human factor, optimizing medicine supply volumes, and minimizing the risks of freezing the enterprise’s working capital in excess inventory.

The research results hold practical value for analytics units and supply departments of retail pharmacy chains, and can also be applied as a decision support tool within operational planning systems.

Keywords: TIME SERIES, PREDICTIVE MODELING, PROPHET MODEL, PYTHON, STREAMLIT, INTERACTIVE DASHBOARD, MAPE METRIC, OPERATIONAL PLANNING, INVENTORY OPTIMIZATION.

ЗМІСТ

Вступ.....	7
1 Стан та аналіз предметної області.....	10
1.1 Опис предметної області	10
1.2 Огляд і аналіз існуючих рішень	13
1.3 Постановка задачі дослідження.....	17
2. Алгоритмічне та інформаційне забезпечення системи	20
2.1 Проектування сучасних інформаційних систем для фармацевтичного ритейлу	20
2.2 Модуль збору та попередньої обробки даних (ETL процес)	22
2.3 Підсистема інженерії ознак (Feature Engineering) для фармацевтичних часових рядів	24
2.4 Архітектура ансамблю моделей прогнозування	26
2.5 Підсистема валідації результатів та бізнес-аналітики.....	28
3 Програмно-технологічне забезпечення системи.....	31
3.1 Реалізація програмного забезпечення	31
3.2 Інтерфейс користувача.....	40
3.3 Тестування розробленої програми	45
Висновки	48
Список використаних джерел	51
Додаток А Лістинг реалізації основних функціональних механізмів програмного модуля	53
Додаток Б Копії опублікованих результатів.....	60

ВСТУП

Актуальність теми. Історично управління асортиментом в аптеках значною мірою залежало від інтуїції та особистого досвіду завідувачів. Ручне формування замовлень несло значні ризики: людський фактор часто призводив до закупівлі надлишкового товару. Як наслідок, усі обігові кошти компанії могли бути "заморожені" в запасах на полицях, що критично знижувало загальну рентабельність та могло навіть поставити під загрозу існування бізнесу. Крім того, неліквідний товар з часом протерміновувався, завдаючи прямих фінансових збитків.

Сьогодні ж фармацевтичний ритейл стрімко рухається в бік автоматизації. Замість ручного втручання впроваджуються програмні рішення, здатні аналізувати попит та самостійно розраховувати оптимальну потребу в медикаментах. Головна мета таких систем – замовляти таку кількість ліків, яка з максимальною ймовірністю реалізується найближчим часом. Це гарантує, що товар не буде довго простоювати на залишках, ризики списання через закінчення терміну придатності зводяться до мінімуму, а вивільнений капітал компанія може направити на більш маржинальні напрямки.

Актуальність цієї кваліфікаційної роботи зумовлена необхідністю створення інтелектуального інструменту для оптимізації аптекних продажів та автоматизації предиктивної аналітики часових рядів фармацевтичної продукції. Для побудови надійного модуля прогнозування у дослідженні застосовується комплексна методологія роботи з даними, що базується на трьох ключових етапах: інженерії ознак та підготовці даних (Feature engineering and data preparation), розвідувальному аналізу часових рядів (Exploratory data analysis / Time-series analysis) та безпосередньому прогнозуванні (Forecasting) із застосуванням алгоритмів машинного навчання. Розроблений програмний модуль перетворює «сирі» щоденні транзакційні дані на дієвий інструмент підтримки прийняття рішень, що забезпечує стабільність бізнес-процесів усієї роздрібної мережі.

Мета дослідження – розробка інтерактивного програмного модуля (MVP-дашборду) для аналізу та предиктивного моделювання щоденних продажів фармацевтичної продукції на основі алгоритмів машинного навчання для оптимізації товарних запасів аптечної мережі.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати особливості предметної області та дослідити існуючі методи аналізу й прогнозування часових рядів у сфері роздрібної торгівлі медикаментами.

- Сформувати конвеєр попередньої обробки та інженерії ознак (*Feature Engineering*) для очищення транзакційних даних від аномалій та врахування екзогенних бізнес-факторів.

- Обґрунтувати вибір та здійснити математичне налаштування предиктивного алгоритму побудови прогнозів (моделі Prophet).

- Спроекувати об'єктно-орієнтовану архітектуру та реалізувати вебзастосунок аналітичного дашборду на базі сучасного технологічного стека (Python, Streamlit, Plotly).

- Провести функціональне тестування розробленого програмного комплексу та оцінити точність прогнозової моделі за допомогою метрик Percentage Error (зокрема MAPE).

Об'єкт дослідження – процеси операційного планування та предиктивного аналізу часових рядів обсягів реалізації товарів у роздрібній торгівлі.

Предмет дослідження – моделі, методи машинного навчання та програмні засоби інтерактивної візуалізації й автоматичного короткострокового прогнозування продажів фармацевтичних препаратів.

Методи дослідження. Для розв'язання поставлених завдань використано: методи об'єктно-орієнтованого аналізу та проєктування програмних систем (UML-моделювання), методи математичної статистики та предиктивного аналізу часових рядів (декомпозиційний адитивний аналіз моделі Prophet), а також методи емпіричного тестування та валідації якості алгоритмів машинного навчання (MAPE, MSE).

Наукова новизна отриманих результатів полягає у вдосконаленні підходу до інтеграції специфічних для українського фармацевтичного ринку календарних екзогенних факторів (фіксовані дні соціальних виплат та специфічні святкові періоди) безпосередньо у внутрішню структуру регресорів адитивної моделі прогнозування часових рядів, що дозволило знизити похибку моделювання динамічного попиту в умовах високої зашумленості даних.

Практичне значення отриманих результатів. Розроблено автономний працездатний програмний модуль у вигляді Streamlit-дашборду, який автоматично обробляє транзакційні масиви даних, що відповідають форматам промислових облікових систем (на кшталт 1C/ERP). Модуль дозволяє автоматизувати розрахунок майбутньої потреби в ліках, наочно візуалізувати плани й прогнози товарообігу та надає бізнес-аналітику інструменти для прийняття рішень, мінімізуючи ризики заморожування обігових коштів підприємства.

Основні положення, висновки і результати дипломної роботи доповідалися, обговорювалися й одержали позитивну оцінку на I Міжнародній науково-практичній конференції студентів і молодих вчених “ICSPM 2026: Intelligent Computing Systems and Project Management / Інтелектуальні обчислювальні системи та управління проєктами” (м. Тернопіль, 2026 р.) (додаток Б).

1 СТАН ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Предметною областю даного дипломного дослідження є процеси інтелектуального аналізу даних та прогнозування обсягів продажу фармацевтичної продукції на мікрорівні локальної аптечної мережі. Для комплексного розуміння проблеми необхідно детально розглянути специфіку фармацевтичного ритейлу, архітектуру збереження торгових даних, метрики ефективності бізнесу (KPI) та математичні виклики, що постають при моделюванні локального попиту.

1.1.1 Специфіка фармацевтичного ритейлу та управління запасами

Фармацевтичний ритейл являє собою складну екосистему, яка кардинально відрізняється від класичної роздрібної торгівлі (FMCG). Головна відмінність полягає в самому продукті: лікарські засоби є товаром першої необхідності із суворими обмеженнями щодо умов зберігання та кінцевих термінів придатності. Згідно з дослідженнями ланцюгів постачання в охороні здоров'я [1], управління фармацевтичними запасами підпорядковується жорсткому принципу FEFO (First Expire, First Out – перший закінчується термін, перший вибуває). Це створює постійний тиск на менеджмент аптек: закупівля надмірної кількості препаратів неминуче призводить до їх протермінування, що генерує прямі фінансові збитки та вимагає витрат на спеціальну утилізацію. З іншого боку, занижений рівень запасів викликає явище дефектури – фізичної відсутності необхідного товару в момент звернення пацієнта. В умовах високої конкуренції (коли аптеки різних мереж часто знаходяться в одному будинку) вартість дефектури вимірюється не лише втраченою поточною маржею, але й втратою лояльності клієнта на довгострокову перспективу. Якщо пацієнт із хронічним захворюванням не знаходить потрібного препарату, він з високою ймовірністю змінить “улюблену” аптеку, що призводить до падіння загального трафіку торгової точки. Отже, головною проблемою предметної області є балансування між ризиком “замороження” обігових коштів у надлишкових товарних запасах та ризиком

втрачених продажів через дефектуру. Історично це завдання покладалося на експертне судження завідувачів аптек, однак з ростом асортименту (який може налічувати десятки тисяч активних позицій – SKU) ручне управління стало неефективним, що зумовило гостру потребу в системах автоматизації та бізнес-аналітики.

1.1.2 Роль бізнес-аналітики та показників ефективності (KPI) в оптимізації маржі

У сучасному ритейлі прийняття рішень базується на даних (Data-Driven Decision Making). Ключовим аспектом предметної області є перехід від простого товарообігу до управління рентабельністю на основі глибокої бізнес-аналітики та розрахунку продуктивних метрик. Головним фінансовим показником ефективності (KPI) роботи аптеки є не просто обсяг виторгу, а валовий прибуток (маржа). Маржинальність різних груп препаратів суттєво відрізняється: життєво необхідні ліки часто мають законодавчо обмежену націнку, тоді як дієтичні добавки, вітаміни або супутня продукція є високомаржинальними. Згідно з методологією управління роздрібними продажами [2], максимізація загальної маржі компанії досягається шляхом забезпечення постійної, безперебійної наявності саме високомаржинальних товарів (маркери прибутковості) та стимулювання їх збуту. У цьому контексті точне прогнозування продажів виступає головним інструментом оптимізації. Математично втрачена маржа через дефектуру розраховується як різниця між прогнозованим попитом та поточним залишком, помножена на маржу з однієї упаковки [2]:

$$Lost_Margin = \max(0, Forecast - Stock) \times M_{unit}. \quad (1.1)$$

Розуміння цієї формули є критичним для предметної області, адже програмний модуль прогнозування створюється не як абстрактний математичний інструмент, а як засіб мінімізації *Lost_Margin*.

Автоматично визначаючи майбутню потребу, система дозволяє відділу закупівель перерозподіляти бюджет, інвестуючи вивільнені кошти в ті товари, які генерують найбільший прибуток для мережі.

1.1.3 Архітектура даних: транзакційні системи, плоскі бази та зведені таблиці

Основою для проведення будь-якого аналізу та побудови моделей машинного навчання у даній предметній області є історичні дані, що генеруються інформаційними системами торгових точок (POS-системами). Кожна операція на касі фіксується у реляційній базі даних аптеки, формуючи гігантські масиви транзакційної інформації. Однак “сирі” дані у вигляді складних багатотабличних структур (де окремо зберігаються довідники товарів, чеки, клієнти, ціни) є незручними для оперативного розвідувального аналізу. Тому стандартом індустрії є трансформація транзакційних логів в аналітичну “плоску базу даних” (flat database) [3]. У такій структурі кожен рядок представляє агреговану інформацію (наприклад, продажі конкретного препарату за конкретну годину з усіма необхідними атрибутами). Використання плоскої бази даних є фундаментальним для розробки гнучких інформаційних систем. Вона дозволяє бізнес-аналітикам швидко генерувати багатовимірні зведені таблиці (pivot tables), які є основним інструментом візуалізації KPI. За допомогою зведених таблиць менеджмент може досліджувати трафік клієнтів (Customer Count) у розрізі годин (Hour) та днів тижня (Day of Week), виявляючи години пік для оптимізації графіків роботи персоналу. Крім того, саме з плоскої бази формуються часові ряди потрібної дискретності (щоденні або щотижневі) для подачі їх на вхід алгоритмам прогнозування.

1.1.4 Специфіка прогнозування часових рядів на мікрорівні

Переходячи до математичної складової предметної області, необхідно відзначити різницю між прогнозуванням на макро- та мікрорівні. На рівні всієї країни чи великого національного дистриб'ютора часові ряди продажів ліків є відносно згладженими. Закон великих чисел компенсує локальні відхилення, що

дозволяє використовувати прості (наївні) моделі прогнозування або класичні середні ковзні [4]. Натомість на мікрорівні (рівні окремої аптеки чи локальної мережі) дані характеризуються високим рівнем дисперсії, нестационарністю та наявністю значного інформаційного шуму (residuals). Продажі в конкретній точці сильно залежать від зовнішніх (екзогенних) факторів. Зниження ціни на препарат (дисконт) може викликати різкий сплеск продажів з подальшим провалом, оскільки постійні клієнти закупили ліки запас. Дні виплати державних пенсій формують стабільні, але вузькі піки попиту на препарати для лікування хронічних захворювань. Державні свята працюють як недільні дні, ламаючи звичну тижневу автокореляцію. Крім того, асортимент налічує десятки тисяч найменувань (SKU), більшість з яких продаються спорадично (1-2 упаковки на тиждень). Побудова точного прогнозу для кожного окремого SKU є математично неможливою задачею через перенавчання (overfitting) моделей на шумі. Тому важливою методологічною особливістю предметної області є агрегація товарів. Аналіз та моделювання доцільно проводити для укрупнених груп – наприклад, для 8 основних категорій фармацевтичних продуктів за міжнародною АТС-класифікацією. Такий підхід дозволяє виділити фундаментальні тренди, згладити випадкові коливання та побудувати надійні моделі прогнозування часових рядів (univariate time series forecasting), здатні враховувати цінові фактори та дні соціальних виплат для проведення “what-if” аналізу під час маркетингових кампаній.

1.2 Огляд і аналіз існуючих рішень

Стрімкий розвиток інформаційних технологій та накопичення великих обсягів даних (Big Data) у роздрібній торгівлі призвели до суттєвої еволюції методів управління запасами. Для обґрунтування архітектури пропонованого програмного модуля необхідно провести критичний аналіз існуючих рішень у сфері прогнозування попиту, починаючи від класичних систем управління запасами до передових алгоритмів машинного навчання.

1.2.1 Еволюція систем управління запасами у фармацевтичному ритейлі

Історично першим і базовим рішенням для управління запасами в аптеках було експертне оцінювання, яке базувалося виключно на інтуїції та досвіді фармацевта або завідувача. З впровадженням перших комп'ютеризованих облікових систем галузь перейшла до використання статичних моделей управління запасами, найпопулярнішою з яких є стратегія "Min-Max" (точка замовлення). Згідно з класичною теорією управління ланцюгами постачання [5], ця система автоматично генерує потребу в товарі лише тоді, коли його залишок на складі опускається нижче заздалегідь встановленого мінімального критичного рівня (Min), замовляючи обсяг, необхідний для досягнення максимального рівня (Max).

Головним недоліком існуючих базових ERP-систем, які досі використовують метод Min-Max, є їхня реактивність. Такі системи фіксують зміну залишку постфактум, але абсолютно не здатні аналізувати динаміку попиту в майбутньому. В умовах фармацевтичного ринку це призводить до "ефекту батога" (bullwhip effect): система може замовити максимальну кількість сезонного противірусного препарату наприкінці епідемії просто тому, що залишок впав нижче мінімуму, що неминуче призведе до протермінування цього товару влітку. Відповідно, виникає потреба в проактивних (предиктивних) рішеннях.

1.2.2 Традиційні статистичні та базові (наївні) моделі прогнозування

Першим кроком до проактивного управління є використання базових математичних моделей. На макрорівні – для великих фармацевтичних виробників або національних дистриб'юторів – найчастіше застосовуються так звані "наївні" моделі (Naïve models). Згідно з методологією аналізу часових рядів [6], наївна модель припускає, що обсяг продажів у наступному періоді буде дорівнювати обсягу продажів у попередньому періоді з додаванням певного фактора макроекономічного росту. Різновидом цього підходу є сезонна наївна модель (Seasonal Naïve), яка копіює дані з аналогічного періоду минулого року.

Хоча такі прості методи прогнозування інтегровані в багато існуючих аналітичних платформ і добре працюють на великих масштабах (де агрегація даних по сотнях аптек згладжує випадкові коливання), вони демонструють критично низьку точність на мікрорівні. Аналіз показує, що дані продажів окремої аптеки характеризуються суттєвою невизначеністю та високою дисперсією. Наївні моделі часто дають збій у разі насичення ринку або при раптових змінах зовнішнього середовища, тому їх використання для щоденного поповнення запасів конкретної аптеки є недоцільним.

1.2.3 Просунуті методи аналізу часових рядів (ARIMA та SARIMA)

Більш зрілим класом рішень, який широко використовується в комерційній аналітиці, є моделі експоненційного згладжування та інтегрованого ковзного середнього (ARIMA). Методологія Бокса-Дженкінса, на якій базується ARIMA, дозволяє знаходити автокореляцію всередині часового ряду, визначаючи залежність поточних продажів від їх історичних значень та залишкових помилок [7]. Для фармацевтичної продукції з чітко вираженими сезонними патернами (наприклад, антиалергійні препарати навесні) застосовується розширена архітектура SARIMA.

Існуючі програмні бібліотеки для побудови моделей ARIMA забезпечують високу академічну точність. Проте їхнє впровадження в повністю автоматизовані конвеєри стикається з обмеженнями: ці алгоритми вимагають суворої стаціонарності часового ряду (постійного середнього значення та дисперсії). Процедури приведення ряду до стаціонарності (диференціювання, логарифмування) ускладнюють інтерпретацію результатів для бізнесу. Крім того, класичні моделі погано справляються з інтеграцією нелінійних екзогенних факторів, таких як знижки чи свята.

1.2.4 Сучасні адитивні моделі бізнес-прогнозування (Prophet)

Для подолання обмежень статистичних методів в індустрії Data Science почали з'являтися спеціалізовані алгоритми для прогнозування бізнес-метрик. Одним із найпотужніших сучасних рішень є модель Prophet, розроблена

дослідниками компанії Facebook у 2017 році [8]. Це адитивна регресійна модель, яка розкладає часовий ряд на компоненти: довгостроковий тренд, багаторівневу сезонність (тижневу та річну) і вплив свят.

Аналіз показує, що рішення на базі Prophet ідеально підходять для ритейлу. На відміну від ARIMA, цей алгоритм стійкий до відсутності даних (missing data), що часто трапляється в аптеках через технічні збої POS-систем або тимчасову відсутність товару. Найбільшою перевагою Prophet є його здатність "з коробки" враховувати вплив подій: дат виплат пенсій, локальних свят та акційних днів, які вносять найбільший хаос у стандартний графік продажів.

1.2.5 Застосування методів глибинного навчання (Deep Learning та LSTM-мережі)

Останнім словом у сфері прогнозування попиту є використання алгоритмів глибинного навчання (Deep Learning), зокрема штучних нейронних мереж. Класичні багат шарові перцептрони (MLP) або базові рекурентні нейронні мережі (RNN) виявилися недостатньо ефективними для обробки довгих послідовностей через проблему "зникаючого градієнта" під час навчання [9].

Вирішенням цієї проблеми стала архітектура мереж довгої короткострокової пам'яті (LSTM – Long Short-Term Memory). Існуючі дослідження доводять, що LSTM-мережі здатні автоматично витягувати найскладніші нелінійні патерни з величезних масивів "сирих" транзакцій. Вони можуть ефективно прогнозувати продажі специфічних груп препаратів, де попит залежить від багатьох неочевидних взаємозв'язків. Перспективними також вважаються гібридні архітектури (CNN-LSTM або ConvLSTM), які поєднують згорткові шари для виділення локальних ознак з рекурентними шарами для аналізу послідовностей. Однак впровадження таких рішень вимагає високих обчислювальних потужностей та ретельної оптимізації гіперпараметрів.

1.2.6 Порівняльний аналіз та обґрунтування вибору ансамблевого підходу

Підсумовуючи аналіз існуючих рішень, можна зробити висновок, що на ринку не існує "срібної кулі" – єдиного алгоритму, який би ідеально прогнозував усі фармакологічні групи препаратів на мікрорівні аптеки.

Готові коробкові продукти (комерційні ERP-системи) найчастіше пропонують надто прості, жорстко заcodedовані методи (Min-Max або ковзні середні), які призводять до дефектури. Хмарні ML-сервіси (наприклад, AWS Forecast) забезпечують високу точність, але є дорогими (proprietary software) та важко інтегруються у специфічні бізнес-процеси локальних мереж з плоскими базами даних.

Найбільш доцільним архітектурним рішенням є розробка власного (in-house) програмного модуля, який використовує концепцію "ансамблю". Комбінування класичних методів (ARIMA) для стабільних товарів, адитивних моделей (Prophet) для товарів із вираженою сезонністю та нейронних мереж (LSTM) для складних нелінійних залежностей дозволить створити гнучку систему ковзного прогнозування (rolling forecast). Такий підхід гарантує мінімізацію похибки (MSE та MAPE) та ефективне використання обчислювальних ресурсів аптечної мережі.

1.3 Постановка задачі дослідження

Головною метою даного дипломного дослідження є розробка та програмна реалізація модуля для аналізу та прогнозування продажів фармацевтичної продукції. Такий модуль має допомогти оптимізувати ключові показники ефективності (KPI) аптечної мережі та покращити процес прийняття управлінських рішень.

У діяльності аптек важливо правильно прогнозувати попит на лікарські засоби, оскільки він може значно змінюватися залежно від сезону, маркетингових акцій, свят або інших факторів. Тому створення інструменту, який дозволяє аналізувати історичні дані продажів та прогнозувати майбутній попит, є актуальним завданням для підвищення ефективності роботи аптечної мережі.

Для досягнення поставленої мети необхідно вирішити ряд взаємопов'язаних завдань, які можна умовно поділити на етапи підготовки даних, аналізу інформації, побудови моделей прогнозування та оцінки отриманих результатів.

У рамках даного дослідження поставлено такі основні завдання:

1. Підготовка та обробка даних

Необхідно здійснити збір історичних даних про продажі лікарських засобів з інформаційних систем аптечної мережі. Після цього виконується попередня обробка даних, яка включає очищення інформації, обробку пропущених значень та виявлення аномальних значень.

На основі підготовлених даних формується структурована база даних, яка дозволяє зручно аналізувати продажі та виконувати подальші аналітичні розрахунки.

2. Аналіз даних та підготовка ознак

На цьому етапі проводиться аналіз отриманих даних для виявлення закономірностей у продажах. Номенклатура лікарських засобів групується у кілька основних фармакологічних категорій відповідно до АТС-класифікації.

Також досліджуються часові ряди продажів для виявлення сезонності, повторюваних тенденцій та інших факторів, які можуть впливати на попит. Додатково враховуються зовнішні фактори, такі як державні свята, дні виплати пенсій, зміни цін та проведення маркетингових акцій.

3. Розробка моделей прогнозування

На основі підготовлених даних розробляються моделі прогнозування продажів. Для цього планується використання різних підходів до аналізу часових рядів, зокрема статистичних моделей та сучасних методів машинного навчання.

Метою цього етапу є створення моделей, які дозволять прогнозувати обсяги продажів як на короткий період часу (для оперативного управління запасами), так і на більш довгострокову перспективу.

4. Оцінка якості прогнозування

Після побудови моделей необхідно оцінити точність отриманих прогнозів. Для цього проводиться порівняння результатів різних моделей за допомогою стандартних показників точності прогнозування.

Отримані результати дозволять визначити найбільш ефективний підхід до прогнозування продажів.

5. Практичне використання результатів

На основі отриманих прогнозів формуються рекомендації щодо управління запасами лікарських засобів у аптеках. Це дозволяє зменшити ризик відсутності необхідних препаратів, оптимізувати складські запаси та підвищити загальну ефективність роботи аптечної мережі.

Реалізація поставлених завдань дозволить створити програмний модуль, який зможе аналізувати історичні дані продажів та прогнозувати майбутній попит на фармацевтичні позиції. Використання подібного інструменту може сприяти підвищенню ефективності управління запасами та покращенню фінансових результатів аптечної мережі.

2. АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Проєктування сучасних інформаційних систем для фармацевтичного ритейлу

Проєктування сучасних інформаційних систем для фармацевтичного ритейлу вимагає створення надійної інфраструктури, здатної обробляти інтенсивні потоки даних (Data-Intensive Applications). Загальна архітектура розробленого програмного комплексу будується за модульним принципом, що забезпечує безперебійний та масштабований обмін інформацією між локальними точками продажу (POS-терміналами) аптек мережі та центральним аналітичним сховищем. Відповідно до передових принципів проєктування систем, що інтенсивно працюють з даними [10], архітектура логічно поділяється на три ключові рівні: рівень джерел даних (Data Source), аналітичне ML-ядро (Processing) та рівень бізнес-логіки (Business Logic / Output) (рисунок 2.1).



Рисунок 2.1 – Візуалізація архітектури модуля

На першому рівні (Data Source) відбувається безперервний збір “сирих” транзакційних даних з усіх аптек. Цей потік включає деталізовану інформацію про кожну одиницю проданого товару. Для забезпечення високої швидкості подальшої обробки та можливості багатовимірного аналізу, ці дані консоліднуються у корпоративному сховищі (Data Warehouse) у форматі оптимізованої “плоскої бази даних” (flat database). Така архітектурна структура є критично важливою: вона усуває необхідність виконання складних SQL-з'єднань (JOIN) під час роботи ML-модуля і дозволяє ефективно готувати масиви інформації для побудови зведених таблиць (pivot tables). Завдяки цьому система може швидко агрегувати дані не лише за цільовими групами препаратів, але й гнучко аналізувати динаміку кількості клієнтів (трафік) у розрізі окремих годин та днів тижня по різних аптеках мережі.

На другому рівні (Processing) функціонує розроблений програмний модуль прогнозування, написаний мовою Python. Він працює як автономний мікросервіс, який за заданим розкладом звертається до “плоскої бази” транзакцій, ініціює конвеєр попередньої обробки даних (Data Pipeline) та проводить обчислення за допомогою ансамблю моделей машинного навчання. Результатом роботи цього рівня є масиви прогнозних значень попиту на майбутні періоди.

Третій рівень (Business Logic) відповідає за інтеграцію отриманих математичних результатів безпосередньо в операційні процеси компанії. Згенеровані прогнози експортуються до центральної ERP-системи мережі. Тут розроблена архітектура вирішує своє головне фінансове завдання: вона підтримує імплементацію методології підвищення загальної маржі компанії. Оскільки система точно прогнозує попит, менеджмент отримує змогу фокусувати закупівлі на цільових категоріях, стимулюючи продажі високомаржинальних товарів. Запобігаючи дефектурі (відсутності на полицях) високомаржинальних препаратів за допомогою предиктивної аналітики, програмний комплекс безпосередньо впливає на зростання загальних показників прибутковості аптечної мережі.

2.2 Модуль збору та попередньої обробки даних(ETL процес)

Фундаментом будь-якої аналітичної системи є якість та структурованість вхідних даних. У випадку аналізу продажів аптечної мережі основним джерелом інформації є транзакційні дані з касових апаратів (POS-систем). Такі дані можуть містити технічні помилки, дублікати записів або пропущені значення, тому перед їх використанням необхідно провести попередню обробку.

Сучасні стандарти побудови систем аналізу даних мовою Python передбачають організацію ізольованих конвеєрів (Data Pipelines) для забезпечення відтворюваності процесів обробки [11]. Розроблений мікросервіс ETL реалізує концепцію послідовної трансформації об'єктів DataFrame, де кожен етап (витягання, Z-очищення викидів, інтерполяція пропусків та агрегація) є логічно відокремленим кроком обчислювального конвеєра, що унеможливорює витік даних (data leakage) під час навчання моделей.

На етапі витягання (Extract) система отримує історичні дані про продажі з інформаційних систем аптечної мережі. До таких даних належать записи чеків, які містять інформацію про дату продажу, товар, кількість та інші параметри транзакції.

Другий етап – перетворення (Transform) – включає процедури очищення та підготовки даних до подальшого аналізу. Реалізація цих операцій здійснюється за допомогою бібліотек мови програмування Python, зокрема Pandas та NumPy, які широко використовуються для роботи з табличними даними. На цьому етапі виконуються такі операції:

Відповідно до поширених підходів аналізу часових рядів, пропущені значення у даних продажів зазвичай не видаляються, оскільки це може порушити безперервність часового ряду [3]. Замість цього застосовуються методи їх заповнення, наприклад інтерполяція або перенесення попереднього значення (forward fill). Такий підхід дозволяє зберегти цілісність даних і забезпечує коректну роботу алгоритмів прогнозування.

Важливою частиною етапу трансформації є також виявлення аномальних значень (outliers). У сфері аптечного ритейлу аномалії можуть виникати через

технічні помилки в системі або через нетипові продажі великої кількості товарів. Для їх виявлення можуть використовуватися статистичні показники, зокрема z-оцінка або міжквартильний розмах. Для автоматичного виявлення аномальних сплесків продажів (наприклад, помилково пробитих чеків або нетипових оптових закупівель) у модулі використовується метод стандартизації або Z-оцінки (Z-score). Алгоритм обчислює відхилення кожного значення продажу від середнього показника по групі за формулою [12]:

$$Z = \frac{x_i - \mu}{\sigma}, \quad (2.1)$$

де x_i - кількість проданих одиниць у конкретний день;

μ - середнє значення продажів за період;

σ – стандартне відхилення.

Якщо абсолютне значення $|Z| > 3$, система класифікує такий продаж як статистичний викид (outlier) і замінює його на згладжене медіанне значення для коректного навчання нейромереж.

Виявлені аномальні значення можуть бути замінені більш типовими показниками, наприклад медіанними значеннями, щоб уникнути спотворення результатів подальшого аналізу.

Після завершення очищення та підготовки даних виконується етап завантаження (Load). На цьому етапі дані структуруються у вигляді єдиної таблиці або так званої плоскої бази даних (flat database). Така структура є зручною для подальшого аналітичного опрацювання та дозволяє швидко виконувати агрегування даних.

Зокрема, на основі сформованої бази можна створювати зведені таблиці (pivot tables), які дозволяють аналізувати продажі у різних розрізах. Наприклад, система може відображати кількість клієнтів у конкретній аптеці залежно від часу доби або дня тижня. Отримані результати можуть використовуватися для

аналізу динаміки продажів, планування запасів товарів, а також оптимізації графіків роботи персоналу аптек.

2.3 Підсистема інженерії ознак (Feature Engineering) для фармацевтичних часових рядів

У задачах машинного навчання якість сформованих ознак (features) часто має більший вплив на точність прогнозової моделі, ніж складність обраного алгоритму. Підсистема інженерії ознак призначена для доповнення базових історичних часових рядів додатковою інформацією, яка допомагає моделям виявляти закономірності у поведінці споживачів. Відповідно до методології підготовки табличних даних для предиктивних моделей [12], процес формування ознак поділяється на два основні етапи: використання внутрішніх показників та інтеграцію зовнішніх факторів (рисунок 2.2).

Першим етапом є агрегація товарної номенклатури. Прогнозування продажів на рівні окремих товарних позицій (SKU) може створювати значний рівень інформаційного шуму через велику кількість позицій. Тому товари об'єднуються у 8 основних макрокатегорій відповідно до міжнародної анатомо-терапевтично-хімічної класифікації (АТС), наприклад препарати для серцево-судинної системи, нервової системи або анальгетики. Такий підхід дозволяє зменшити варіативність даних та більш чітко визначити основні тенденції продажів.

Другим етапом є формування часових ознак. На основі дати транзакції автоматично визначаються додаткові параметри: день тижня, номер місяця, номер тижня у році, а також ознаки, що вказують на вихідні дні. Це дає змогу алгоритмам враховувати сезонні коливання попиту.

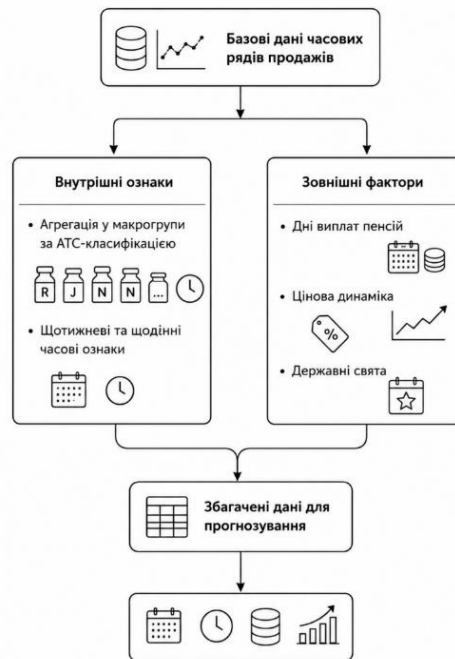


Рисунок 2.2 – Візуалізація внутрішнього і зовнішнього впливу на багатші дані

Важливим елементом підвищення точності прогнозу є врахування зовнішніх факторів, що впливають на діяльність аптечної мережі. До набору даних додаються такі змінні:

Дні виплати пенсій та соціальних допомог. У ці періоди часто спостерігається підвищений попит на лікарські засоби, особливо препарати для постійного використання. Це пов'язано з тим, що значна частина покупців здійснює закупівлі саме після отримання виплат.

Цінова динаміка та акційні пропозиції. Зміни цін або проведення маркетингових акцій можуть суттєво впливати на обсяги продажів. Врахування цього фактору дозволяє відрізнити природні зміни попиту від короткострокових коливань, викликаних знижками.

Державні свята. У святкові дні змінюється стандартний режим роботи та поведінка покупців, що може впливати на структуру продажів. Тому ці дні позначаються у даних окремими ознаками, що дозволяє моделі коректніше враховувати сезонні коливання.

2.4 Архітектура ансамблю моделей прогнозування

Основним "мозком" розробленої програми є блок прогнозування. Оскільки ми аналізуємо 8 різних груп ліків, їхні продажі поведуться зовсім по-різному: одні препарати купують стабільно кожного дня, інші залежать від сезону, а треті продаються стрибками через акції. Очевидно, що один універсальний алгоритм не зможе однаково добре передбачити попит на всі ці товари. Тому в програмі реалізовано підхід "ансамблю": система паралельно запускає три різні моделі, порівнює їхню точність і для кожної групи ліків обирає ту, яка впоралася найкраще.

Важливою фішкою нашої архітектури є використання так званого "ковзного прогнозу" (rolling forecast). Програма не робить один статичний прогноз на півроку вперед. Натомість вона постійно оновлює свої передбачення (наприклад, на найближчі 7–14 днів) щоразу, як тільки з кас аптеки в базу потрапляють нові чеки. Це дозволяє системі швидко реагувати на реальні зміни попиту і вчасно формувати замовлення для поповнення запасів.

У нашому ансамблі працюють три основні алгоритми:

1. Класичні статистичні моделі (ARIMA/SARIMA). Вони найкраще підходять для ліків зі стабільними продажами. Суть алгоритму ARIMA досить проста [13]: він бере історію продажів і шукає математичну залежність поточного попиту від того, що продавалося вчора, позавчора чи тиждень тому. Якщо в продажах є чітка сезонність, програма використовує розширену версію моделі – SARIMA.

2. Адитивна модель Prophet [14]. Цей алгоритм, розроблений інженерами Facebook, ідеально підходить для товарів, продажі яких сильно стрибають через вихідні або свята. Prophet розбиває історію продажів на три зрозумілі складові: загальний тренд (ростуть продажі чи падають), сезонність (дні тижня, місяці) та окремі події (наприклад, дні виплат пенсій чи державні свята). Перевага Prophet у тому, що він не "ламається", якщо в даних є пропуски, і дуже швидко адаптується до нових трендів.

Математично алгоритм Prophet розкладає часовий ряд продажів на кілька незалежних компонентів, що дозволяє гнучко моделювати складну поведінку покупців. Базове рівняння [8] прогнозу має вигляд адитивної регресії:

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

де $g(t)$ – функція тренду (відображає довгострокове зростання або падіння продажів ліків);

$s(t)$ – функція сезонності (моделює тижневі або річні цикли попиту);

$h(t)$ – функція подій та свят (враховує дні виплат пенсій, державні вихідні та періоди знижок);

ϵ_t – нормально розподілена помилка моделі (випадковий шум).

3. Нейронні мережі (LSTM). Для найскладніших випадків, де залежності неочевидні, ми використовуємо рекурентні нейронні мережі довгої короткострокової пам'яті (LSTM). Проблема звичайних нейромереж у тому, що коли вони аналізують довгі періоди (наприклад, продажі за два роки), вони часто "забувають" стару інформацію. В науці це називається проблемою зникаючого градієнта [15]. Мережі LSTM мають спеціальні "вентилі", які вирішують: яку інформацію варто запам'ятати надовго (наприклад, щорічні зимові спалахи застуди), а яку краще забути як звичайний випадковий шум. Це дозволяє моделі знаходити дуже складні закономірності в поведінці покупців. Як доведено у дослідженні Хапсарі [16], пряме прогнозування зашумлених часових рядів на рівні окремих фармацевтичних позицій (SKU) суттєво знижує точність через спорадичність попиту. Автори обґрунтовують, що попереднє групування номенклатури за кластерами або категоріями (аналогічно до використаної у роботі АТС-класифікації) з подальшим навчанням рекурентних нейромереж (LSTM/Bi-LSTM) дозволяє моделі ефективніше захоплювати довгострокові часові залежності та адекватно реагувати на нелінійні флуктуації попиту. Проганяючи дані через усі три варіанти, наш програмний модуль гарантує, що

аптека отримає найточніший розрахунок потреби для кожної конкретної групи препаратів..

2.5 Підсистема валідації результатів та бізнес-аналітики

Для того щоб розроблений програмний модуль мав реальну практичну цінність для аптечної мережі, він повинен не лише генерувати математичні прогнози, але й надавати інструменти для оцінки їхньої надійності. Тому фінальним компонентом архітектури є підсистема валідації та перетворення даних у бізнес-метрики.

Механізм перевірки точності моделей: у процесі роботи програма автоматично тестує навчені алгоритми (ARIMA, Prophet, LSTM) на так званій відкладеній (тестовій) вибірці. Тобто система приховує від моделі частину найновіших історичних даних (наприклад, продажі за останній місяць), змушує її зробити прогноз на цей період, а потім накладає отриманий результат на реальні чеки.

Для кількісної оцінки якості прогнозування модуль розраховує дві ключові метрики [6]:

1. Середня абсолютна відсоткова помилка (MAPE): Це основний індикатор для бізнес-користувачів (завідувачів аптек або категорійних менеджерів). Вона показує, на скільки відсотків у середньому помиляється система. Відсотки є найбільш універсальною і зрозумілою мірою для оцінки ризиків при закупівлі.

2. Середньоквадратична помилка (MSE): Ця метрика має критичне значення для оптимізації алгоритму. Оскільки вона зводить помилки у квадрат, система отримує великий "штраф" за значні відхилення. Для аптеки краще, щоб програма частіше помилялася на 1-2 упаковки дешевого препарату, ніж один раз не передбачила різкий стрибок попиту на 100 упаковок життєво необхідних ліків. Алгоритм, який має найменший показник MSE, автоматично обирається як найбільш безпечний для формування замовлення.

Для програмного розрахунку якості навчених моделей на тестовій вибірці застосовуються дві ключові математичні метрики. Середня абсолютна відсоткова помилка (MAPE) розраховується за формулою [4]:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|, \quad (2.3)$$

де A_t – фактичний обсяг продажів;

F_t – прогнозований обсяг;

n – кількість днів.

Для жорсткого штрафування моделі за критичні промахи (наприклад, великі недозамовлення) використовується середньоквадратична помилка [4]:

$$MSE = \frac{1}{n} \sum_{t=1}^n (A_t - F_t)^2. \quad (2.4)$$

Окрім технічних метрик, підсистема бізнес-аналітики конвертує результати прогнозування у фінансові KPI. Зокрема, алгоритм оцінює потенціал збереження маржинальності компанії через уникнення дефектури (втрачених продажів) за формулою [5]:

$$Lost_Margin = \sum_{t=1}^n \max(0, F_t - S_t) \times M_{unit}, \quad (2.5)$$

де F_t – прогнозований попит;

S_t – поточний фактичний залишок препарату в аптеці (Stock);

M_{unit} – маржа з однієї упаковки препарату.

Ця математична логіка дозволяє менеджменту приймати рішення на основі даних (Data-Driven), фокусуючись на забезпеченні аптек товарами, які генерують найбільший прибуток.

Інтеграція в бізнес-процеси та вплив на KPI: головне завдання підсистеми бізнес-аналітики – конвертувати точність алгоритмів у фінансовий прибуток. Згідно з сучасними підходами до впровадження Data Science у ритейлі [2], результати роботи моделі використовуються для розрахунку оптимального рівня товарного запасу.

Розроблений модуль дозволяє менеджменту мережі вирішити дві головні проблеми:

- Мінімізація дефектури: система заздалегідь сигналізує про необхідність закупівлі цільових, високомаржинальних препаратів, гарантуючи, що вони завжди будуть на полицях під час пікового попиту. Завдяки збільшенню частки продажів таких товарів зростає загальна маржинальність компанії.

- Оптимізація обігових коштів: відмова від закупівель “наосліп” або ручного замовлення дозволяє уникнути перезатарювання аптек. Компанія перестає заморожувати гроші в неліквідних запасах і кардинально знижує ризики протермінування ліків.

У підсумку, програмний комплекс функціонує як повноцінний аналітичний інструмент, що перетворює “сирі” дані чеків на конкретні рекомендації із закупівель, підвищуючи загальну операційну ефективність бізнесу.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Реалізація програмного забезпечення

3.1.1 Структурна організація та архітектура системи

Для програмної реалізації інформаційної системи прогнозування продажів фармацевтичної продукції обрано класичну трирівневу архітектуру (Three-Tier Architecture), що опирається на модульний принцип проєктування. На рівні даних реалізовано автоматизований конвеєр зчитування транзакцій `sales_data.csv` та плану `plan_data.csv`, логіка якого закладена у функції `load_data` (див. Додаток А).

Цей підхід забезпечує високу масштабованість, автономність компонентів та полегшує подальше супроводження і розвиток програмного продукту. Структура розробленого модуля проєктувалася у вигляді трьох ізольованих, але взаємопов'язаних рівнів (шарів): шару даних (Data Layer), шару бізнес-логіки (Business Logic Layer) та шару представлення (Presentation Layer).

Шар даних (Data Layer) – цей рівень є фундаментом системи і відповідає за збереження, цілісність та постачання вхідної інформації для подальшого аналізу. Оскільки в рамках даної роботи розробляється прототип (MVP) аналітичного інструменту, пряма інтеграція з промисловими реляційними базами даних на цьому етапі була замінена на роботу з локальними сховищами.

Джерелом інформації виступають звичайні файли форматів CSV та Excel, які містять вивантаження транзакцій, чеків та обсягів реалізації препаратів у розрізі окремих аптек. Шар даних забезпечує швидке зчитування цих масивів для передачі на наступний рівень.

Шар бізнес-логіки (Business Logic Layer) – обчислювальне ядро системи, у якому зосереджена основна математична та алгоритмічна складова. Цей рівень концептуально розділений на два підмодулі:

Модуль попередньої обробки (Data Preprocessing): відповідає за підготовку «сирих» транзакційних даних до формату, придатного для машинного навчання. Тут виконується очищення даних від аномалій (викидів), заповнення пропущених значень (NaN) методами інтерполяції, а також часова агрегація

продажів (поденна або потижнева). Процес реалізовано за допомогою бібліотек Pandas та NumPy.

Модуль прогнозування (Forecasting Engine): центральний блок прийняття рішень на основі алгоритму Prophet. Він приймає підготовлені часові ряди, проводить декомпозицію даних, виявляючи нелінійні тренди, а також тижневу та річну сезонність, що є критично важливим для фармацевтичного ринку (наприклад, зростання попиту на противірусні препарати в осінньо-зимовий період, тобто в найбільший сезон продажів).

Шар представлення (Presentation Layer) – рівень, що відповідає за безпосередню взаємодію з кінцевим користувачем та візуалізацію обчислених результатів. Він реалізований як єдиний веб-додаток на базі фреймворку Streamlit. Цей шар приймає керуючі команди від користувача (вибір горизонту прогнозу, фільтрація за АТС-групами чи конкретними аптеками), ініціює перерахунок моделі на рівні бізнес-логіки та повертає результат у вигляді інтерактивних графіків і розрахованих метрик (MAPE, MSE).

Взаємодія між усіма описаними шарами реалізована за принципом послідовного конвеєра даних (Data Pipeline). Потік інформації рухається односпрямовано: від завантаження файлу на шарі даних, через трансформацію та математичне моделювання на шарі бізнес-логіки, до фінального рендерингу на шарі представлення. Зворотній зв'язок забезпечується через інтерактивні елементи керування: будь-яка зміна параметрів на шарі представлення автоматично запускає новий цикл конвеєра. Логічну взаємодію та зв'язки між програмними компонентами проілюстровано на загальній структурній схемі програмної системи (рисунок 3.1).

3.1.2 Об'єктно-орієнтована модель та програмні компоненти

Незважаючи на те, що фреймворк Streamlit передбачає декларативний та багато в чому процедурний підхід до побудови інтерфейсів, обчислювальне ядро та бізнес-логіка інформаційної системи були спроектовані з використанням логічної парадигми об'єктно-орієнтованого програмування (ООП). Програма складається з чотирьох класів, кожен із яких робить свою конкретну роботу:

– Один клас відповідає за завантаження та первинну обробку даних.

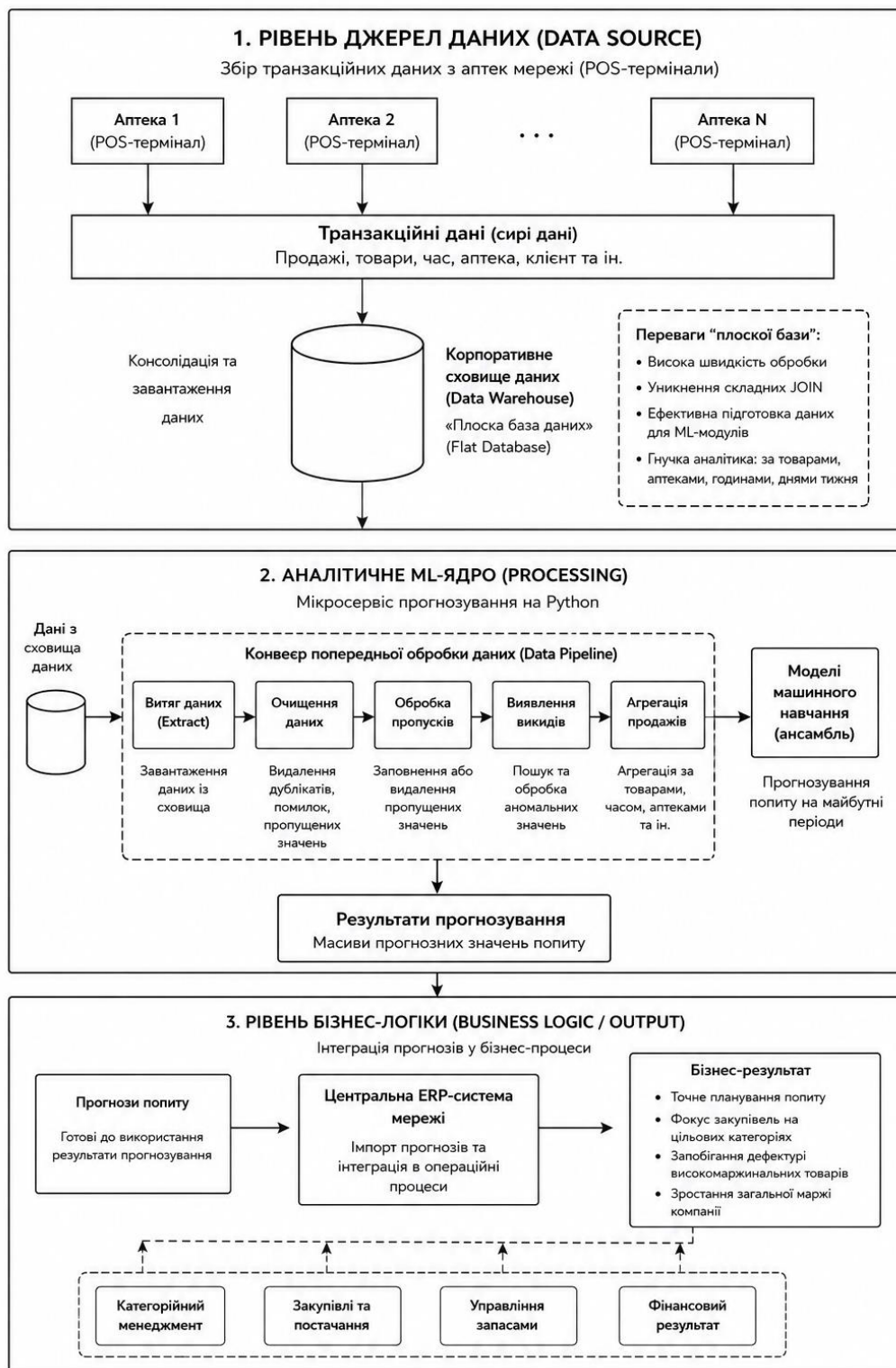


Рисунок 3.1 – Структурна схема програмної системи

- Другий – за розрахунки та бізнес-логіку.
- Третій – за збереження або передачу результатів.

– Четвертий – за відображення в інтерфейсі.

Такий поділ дозволяє легко знайти потрібний код, виправити баг в одному місці не зачіпаючи решту, і при потребі замінити будь-який блок без переписування всього проєкту:

Клас `DataLoader` (Завантажувач даних). Цей компонент повністю відповідає за низькорівневу взаємодію з файловою системою та первинний парсинг історичних транзакцій аптечної мережі.

Атрибути: `raw_data` (зберігає початковий «сирий» масив даних у вигляді об'єкта `DataFrame`), `processed_data` (зберігає очищений масив після первинної обробки).

Ключові методи: `LoadCSV()` – виконує безпечне зчитування файлів із перевіркою кодувань та десеріалізацією форматів дат; у реалізованій програмі (Додаток А) цей метод представлений безпечним читанням факту продажів та інтеграцією планових показників через бібліотеку `Pandas`. `CleanData()` – реалізує логіку обробки пропущених значень (`NaN`), заповнення порожнеч у планах та кодування бізнес-фіч на основі календаря державних свят. `AggregateSales()` – виконує групування транзакцій за днями й обраними категоріями препаратів за допомогою внутрішніх інструментів агрегації даних.

Клас `DataPipeline` (Конвеєр підготовки даних). Виступає як проміжний адаптер (шлюз) між модулем завантаження даних та безпосереднім алгоритмом машинного навчання. Атрибути: `is_ready` (логічний прапорець, що сигналізує про успішне завершення трансформації даних та готовність до навчання).

Ключові методи: `PrepareForProphet()` – здійснює специфічне форматування масиву. Оскільки математична бібліотека `Prophet` вимагає суворого найменування вхідних векторів, цей метод трансформує колонку з тимчасовими мітками на `ds`, а колонку з обсягами реалізації – на `y`. `MergeForecastWithHistory()` – виконує об'єднання історичних часових рядів із розрахованими прогностичними значеннями, а також реалізує логіку «містка» (`link_row`) для безперервності ліній тренду перед передачею на шар рендерингу.

Клас `ForecastingModel` [17] (Модель прогнозування). Центральний об'єкт системи, що інкапсулює математичний апарат, налаштування гіперпараметрів та

логіку роботи алгоритму Prophet. Атрибути: `prophet_model` (екземпляр базового класу Prophet), `forecast_results` (таблиця з результатами майбутніх передбачень та розрахованими довірчими інтервалами). Ключові методи: `TrainModel()` – ініціює процес машинного навчання на підготовленому історичному ряді з обов'язковим додаванням та навчанням зовнішніх екзогенних регресорів святкових днів (`Is_Holiday`) та періодів соціальних виплат (`Is_Payday`). `MakePrediction()` – будує майбутній часовий фреймворк на заданий користувачем горизонт планування та обчислює очікуваний попит. `CalculateMAPE()` та `CalculateMSE()` [18] – аналітичні методи для оцінки якості роботи математичного ядра шляхом розрахунку середньої абсолютної відсоткової похибки та середньоквадратичної помилки відхилень.

Клас `DashboardManager` (Менеджер інтерфейсу). Відповідає за ініціалізацію вебсесії, динамічне управління станами віджетів Streamlit та фінальний рендеринг графічних елементів.

Атрибути: `selected_horizon` (зберігає обраний користувачем горизонт прогнозування), `current_dataset` (поточний активний набір даних).

Ключові методи: `RenderSidebar()` – побудова та обробка подій лівої бічної панелі управління (віджети вибору АТС-груп, діапазону дат, деталізації графіка та запуску обчислень). `DisplayMetrics()` – рендерить верхній інформаційний блок, виводячи на екран картки ключових бізнес-показників (виконання плану, похибка MAPE, прогноз в упаковках та товарообіг).

`PlotForecast()` – відповідає за побудову складного інтерактивного лінійного графіка План-Факт на основі об'єктів візуалізації Plotly [19].

Повний лістинг реалізації зазначених функціональних механізмів, що відображають взаємодію цих компонентів, наведено у Додатку А.

Логічну структуру об'єктно-орієнтованої моделі системи, взаємозв'язки та типи залежностей між переліченими класами графічно зображено за допомогою UML-діаграми класів (рисунок 3.2).

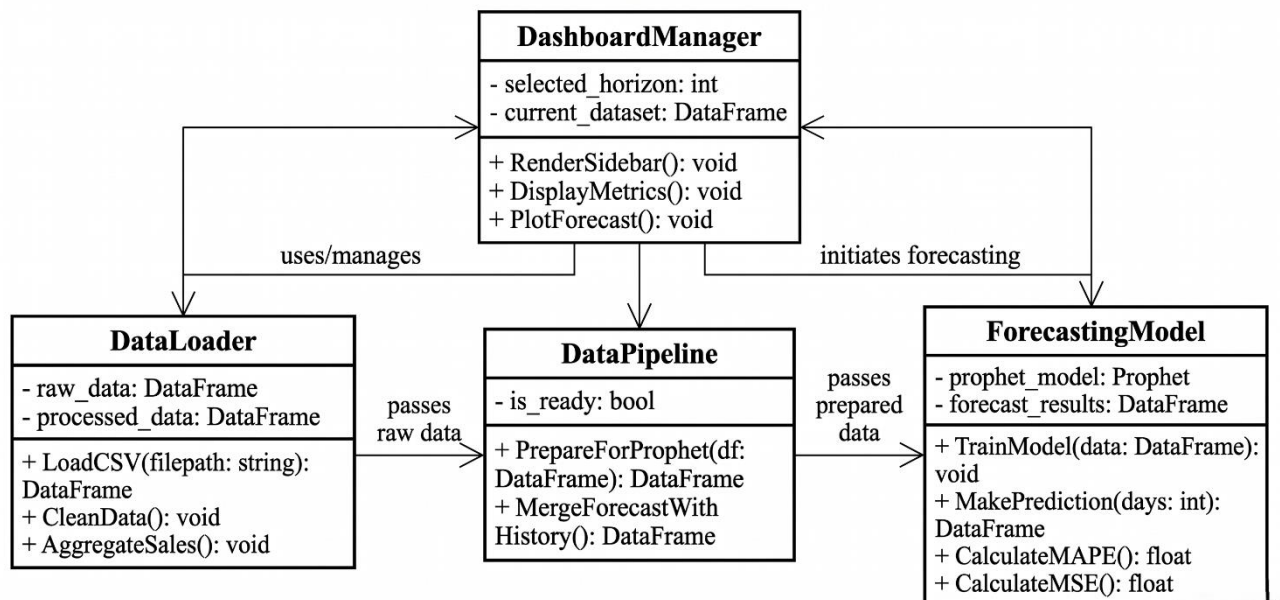


Рисунок 3.2 – UML-діаграма класів програмної системи прогнозування

3.1.3 Життєвий цикл та динаміка станів програми

Окрім статичної об'єктно-орієнтованої структури, важливою складовою проєктування інформаційної системи є визначення її динамічної поведінки в часі. Оскільки вебдодаток побудований на базі фреймворку Streamlit, він використовує реактивну модель виконання: будь-яка зміна вхідних параметрів ініціює повний або частковий перезапуск скрипта з оновленням відповідних станів.

Життєвий цикл розробленого MVP-дашборду можна описати як скінченний автомат (State Machine), що проходить через низку чітко визначених станів від моменту запуску до виведення фінальної аналітики. Динаміку поведінки графічного інтерфейсу та переходи між цими режимами проілюстровано за допомогою UML-діаграми станів (рисунок 3.3).

Процес функціонування системи охоплює такі основні стани:

1. Ініціалізація та завантаження даних (Initialization State). Це початковий стан системи відразу після запуску вебсервера. На цьому етапі програма автоматично звертається до кореневої директорії проєкту та виконує фонове зчитування масивів історичних транзакцій аптечної мережі. Тригером для переходу в наступний стан є успішне завершення імпорту файлів у пам'ять (створення об'єктів DataFrame).

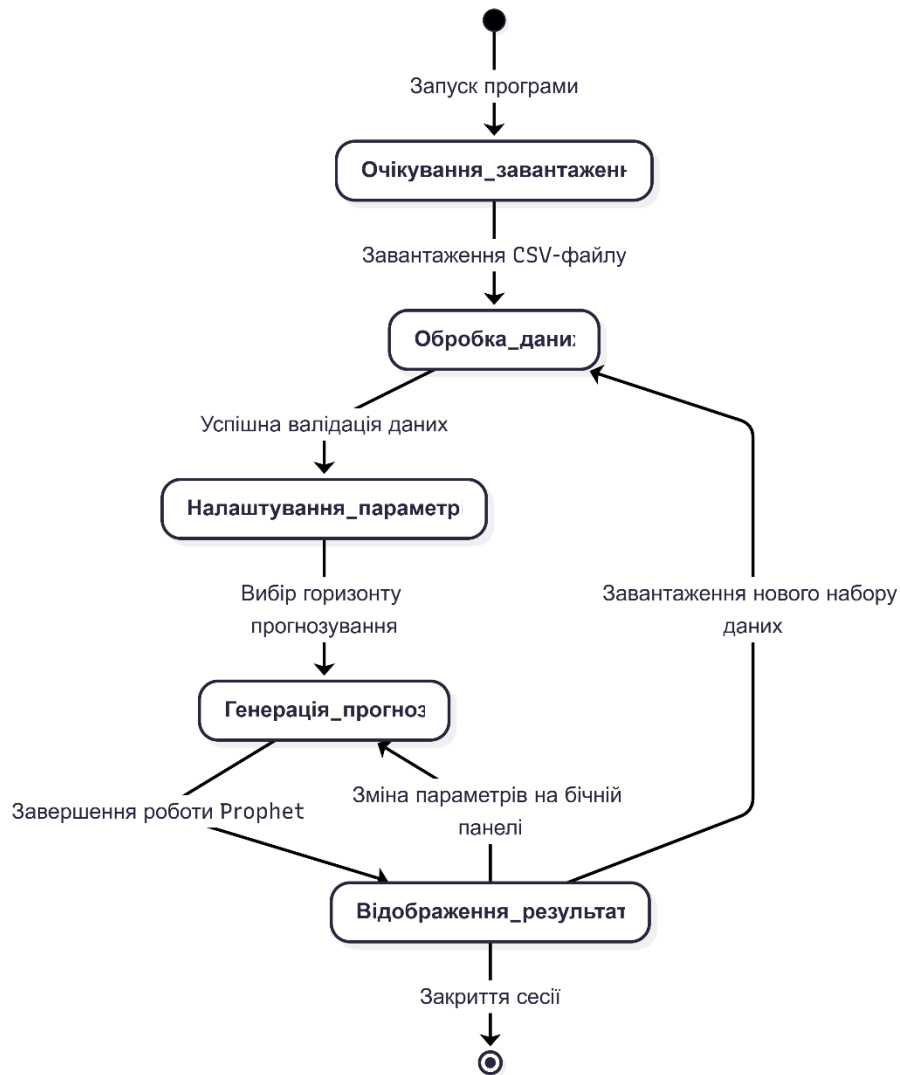


Рисунок 3.3 – UML діаграма станів графічного інтерфейсу дашборду

2. Попередня обробка (Processing State). Система переходить у режим валідації та структурування завантажених даних. Виконується очищення від пропущених значень та агрегація. Якщо на цьому етапі виявляються критичні пошкодження структури файлу, система може перейти в стан помилки. При успішній валідації відбувається перехід до очікування команд користувача.

3. Налаштування параметрів (Configuration State). У цьому стані система перебуває в режимі очікування (Idle). Програма вивела на екран бічну панель інструментів і чекає, поки аналітик або менеджер встановить необхідні параметри: обере цільову АТС-групу препаратів, визначить рівень просторової деталізації (конкретна аптека чи вся мережа) та задасть горизонт прогнозування. Зміна будь-якого параметра виступає тригером, що переводить систему в активний обчислювальний стан.

4. Генерація прогнозу (Forecasting State) Активний стан математичних обчислень. Система ініціює екземпляр класу `ForecastingModel`, передає йому відфільтрований часовий ряд і запускає процес машинного навчання алгоритму `Prophet`. У цей час інтерфейс користувача блокується інформаційними індикаторами завантаження (спінерами), щоб запобігти паралельним несумісним запитам. Тригером виходу з цього стану є завершення розрахунку майбутніх значень та метрик помилок.

5. Відображення результатів (Result State) Фінальний стан циклу, у якому обчислювальне ядро передає результати на шар представлення. Система рендерить інтерактивний графік прогнозу, виводить таблицю даних та розраховані значення MAPE і MSE.

Особливістю реактивної архітектури є те, що стан «Відображення результатів» не є тупиковим. Якщо користувач вирішить змінити будь-який повзунок або фільтр на бічній панелі (наприклад, змінити горизонт з 14 на 30 днів), система миттєво згенерує подію (Event), яка переведе програму назад у стан «Генерація прогнозу» для перерахунку даних. Це забезпечує безперервну інтерактивну взаємодію без необхідності перезавантаження всієї сторінки.

3.1.4 Обґрунтування вибору технологічного стека

Вибір програмних засобів та технологічного стека є критично важливим етапом проєктування інформаційної системи, оскільки він безпосередньо впливає на швидкість розробки, надійність обчислень та гнучкість подальшого масштабування. Зважаючи на те, що метою роботи є створення MVP-модуля (Minimum Viable Product) для аналітики часових рядів, технологічний стек формувався з акцентом на ефективність роботи з даними та швидкість побудови користувацького інтерфейсу.

Мова програмування та екосистема. Основною мовою програмування для реалізації бекенду та обчислювального ядра було обрано Python. На сьогодні Python є галузевим стандартом (де-факто) у сфері Data Science, машинного навчання та предиктивної аналітики. Вибір цієї мови зумовлений її лаконічним

синтаксисом, високою швидкістю написання коду та, найголовніше, наявністю величезної екосистеми спеціалізованих бібліотек.

Шар обробки даних. Для реалізації конвеєра підготовки даних (Data Preprocessing) використано бібліотеки Pandas та NumPy. Інструментарій Pandas [20] ідеально підходить для обробки транзакційних вивантажень: він дозволяє ефективно виконувати агрегацію продажів, злиття таблиць, фільтрацію викидів та обробку пропущених значень. Використання векторизованих операцій під капотом цих бібліотек (на базі мови C) забезпечує високу продуктивність навіть при роботі з великими масивами історичних чеків аптечної мережі.

Алгоритм прогнозування. Як основне математичне ядро системи було обрано бібліотеку Prophet (розроблену дослідницькою групою Meta Core Data Science). Вибір Prophet обґрунтовується специфікою предметної області – фармацевтичними продажами. На відміну від класичних статистичних моделей (наприклад, ARIMA), які вимагають складного ручного налаштування стаціонарності, або глибоких нейромереж (LSTM), які потребують величезних обсягів даних і тривалого часу на навчання, Prophet створений спеціально для бізнес-аналітики. Його головні переваги для даної системи:

Гнучка робота з множинною сезонністю: алгоритм чудово розпізнає як тижневі коливання (наприклад, падіння продажів у вихідні), так і річну сезонність (зростання попиту на певні ліки восени та взимку).

Стійкість до аномалій: Prophet автоматично згладжує вплив поодиноких статистичних викидів (наприклад, гуртової закупівлі препарату одним клієнтом) та дозволяє враховувати вплив святкових днів на загальний попит.

Швидкість: навчання моделі на історичних даних однієї аптеки займає лічені секунди, що дозволяє використовувати його в інтерактивних вебдодатках у режимі реального часу.

Шар представлення (веб-інтерфейс). Для реалізації графічного інтерфейсу користувача було використано сучасний фреймворк Streamlit [21]. У класичному підході розробка подібного дашборду вимагала б створення окремого бекенду (наприклад, на Django або FastAPI) та фронтенду (із застосуванням HTML, CSS та фреймворків JavaScript типу React чи Vue).

Застосування фреймворку Streamlit дозволяє успішно реалізувати парадигму швидкої розробки аналітичних додатків [22], нівелюючи надмірну складність проєктування інтерфейсів шляхом безпосередньої трансформації інструментальних скриптів Python у повноцінні інтерактивні вебзастосунки. Завдяки нативній інтеграції з датафреймами Pandas та профільними бібліотеками візуалізації, цей інструмент забезпечує можливість повністю сфокусуватися на реалізації внутрішньої бізнес-логіки та обчислювальних алгоритмів математичної моделі. Окрім цього, декларативний підхід до рендерингу графічних елементів та вбудована підтримка реактивного управління станами (State Machine) дозволяють програмі функціонувати у форматі односторінкового вебзастосунку (SPA). Таке архітектурне рішення дозволяє повністю уникнути проєктування громіздкої класичної клієнт-серверної інфраструктури (із залученням фронтенд-фреймворків JavaScript та розробкою відокремленого REST API для бекенду) на етапі створення працюючого прототипу (MVP), гарантуючи при цьому сучасний, високореактивний та ергономічний інтерфейс користувача. Поєднання Python, Pandas, Prophet та Streamlit утворює цілісний, високопродуктивний стек, який ідеально відповідає вимогам швидкої розробки надійних аналітичних інструментів для ритейлу.

3.2 Інтерфейс користувача

3.2.1 Принципи побудови та ергономіка вебдодатка

Проектування інтерфейсу користувача базувалося на принципах мінімізації рутинних дій, інтуїтивної зрозумілості та максимальної інформативності. Головним завданням під час розробки MVP-дашборду було створення зручного інструменту для швидкої оцінки майбутнього попиту на медикаменти без необхідності написання програмного коду чи безпосередньої роботи з базами даних.

Інтерфейс реалізовано у вигляді єдиного вебдодатка (Single-Page Application) за допомогою бібліотеки Streamlit. Такий підхід дозволив уникнути

складних багаторівневих меню, які часто перевантажують інтерфейс, та зосередити увагу користувача виключно на аналітичному процесі.

Важливою ергономічною особливістю поточної реалізації розробленого модуля є автоматичне фонове завантаження масивів історичних даних. Наразі система налаштована таким чином, що актуальний файл із транзакціями підтягується безпосередньо з кореневої директорії проєкту при запуску програми. Це позбавляє користувача необхідності щоразу вручну шукати, обирати та імпортувати вхідні файли, що значно пришвидшує початок роботи з дашбордом та мінімізує ймовірність помилок при завантаженні неформатованих даних.

Структурно графічний інтерфейс програми поділяється на дві основні логічні зони: бічну панель налаштувань (Sidebar) та головну робочу область (Main Workspace). Таке компонування забезпечує чітке просторове розмежування інструментів керування моделлю та блоку візуалізації результатів обчислень.

3.2.2 Панель управління та налаштування параметрів моделі

Бічна панель (Sidebar) у розробленому вебдодатку виконує функцію головного пульта управління аналітичним процесом. Вона містить згруповані інтерактивні віджети, які дозволяють користувачеві формувати точний запит до математичної моделі прогнозування без необхідності написання програмного коду чи SQL-запитів.

Інтерфейс бічної панелі, що має назву «Налаштування», логічно поділений на такі послідовні елементи управління:

Випадаючий список «Категорія АТС»: Ключовий фільтр асортименту, який дозволяє користувачеві обрати цільову фармакологічну групу препаратів (наприклад, «C01 - Кардіологічні») для оцінки її трендів та побудови прогнозу.

Блок вибору дат «Історичний період»: Текстово-календарний віджет, що задає часові межі вхідних (навчальних) даних. Користувач може чітко обмежити масив транзакцій (наприклад, з 01.01.2025 по 31.12.2025), на якому буде навчатися алгоритм машинного навчання, відсікаючи застарілі або нерелевантні дані.

Випадаючий список «Деталізація графіка»: Селектор частоти агрегації часового ряду. Цей інструмент дозволяє аналітику гнучко налаштовувати крок візуалізації та обчислень (наприклад, агрегація «По місяцях» для виявлення глобальних макротрендів, або потижнево для точнішого операційного контролю).

Випадаючий список «Горизонт прогнозу»: Надає можливість вказати конкретний часовий проміжок у майбутньому (наприклад, 1 місяць), на який система має екстраполювати історичні дані та розрахувати очікуваний попит.

Кнопка ініціалізації «Згенерувати прогноз»: Важливий елемент управління, що виступає тригером для старту обчислень. На відміну від повністю реактивного оновлення інтерфейсу, наявність окремої кнопки дозволяє користувачеві спочатку встановити всі необхідні параметри фільтрації, а вже потім ініціювати ресурсомісткий процес навчання моделі Prophet. Це суттєво оптимізує навантаження на обчислювальне ядро системи.

3.2.3 Візуалізація результатів та бізнес-аналітика

Головна робоча область вебдодатка призначена для репрезентації результатів роботи алгоритму машинного навчання та надання користувачеві інструментів для прийняття управлінських рішень. Після ініціалізації обчислень на екран виводиться сформований аналітичний звіт, який візуально та логічно поділений на три ключові блоки:

1. Блок зведених показників (KPI Metrics) [23]. У верхній частині робочої області розташовані інформаційні картки, які акумулюють загальні результати обчислень для обраного часового періоду та категорії товарів. У поточній реалізації дашборду виводяться чотири ключові бізнес-метрики: Виконання плану: відсотковий або кількісний показник, який демонструє, наскільки фактичні продажі за обраний історичний період відповідали встановленим плановим значенням.

Похибка прогнозу (MAPE): середня абсолютна відсоткова похибка (Mean Absolute Percentage Error). Це критично важлива метрика якості роботи моделі Prophet, яка показує середній відсоток відхилення прогнозу від фактичних даних.

Майбутні продажі (упаковки): сумарна прогнозована кількість реалізованих одиниць товару на заданий користувачем горизонт прогнозування.

Прогноз ТО (Товарообігу): очікуваний фінансовий результат у грошовому еквіваленті, розрахований на основі прогнозованих кількісних продажів. Ця метрика дозволяє менеджменту оцінювати майбутні грошові потоки та планувати бюджет закупівель.

2. Інтерактивний графік часового ряду. Центральним елементом аналітики є графік прогнозу продажів, реалізований за допомогою бібліотеки Plotly. Графік є повністю інтерактивним: він підтримує динамічне масштабування осей, виділення окремих фрагментів (Zoom) та відображення точних числових значень при наведенні курсора на конкретну точку (Tooltips). Для забезпечення високої інформативності графік поєднує кілька візуальних шарів:

- Фактичні історичні продажі відображаються у вигляді маркерів або ліній.
- Згенерований прогноз на майбутні періоди представлено окремою лінією тренду.

Для зручної навігації на графік нанесено вертикальну розділову лінію з текстовим маркером «← Історія | ПРОГНОЗ →», яка чітко розмежовує фактичні минулі транзакції та результати екстраполяції машинного навчання.

3. Деталізоване табличне представлення (Dataframe View). Під графічним блоком результати прогнозування дублюються у вигляді структурованої таблиці даних. Таблиця містить деталізовану інформацію з розбивкою по датах (або періодах), об'єднуючи колонки з фактичними обсягами та прогнозованими значеннями. Вбудовані можливості інтерфейсу Streamlit дозволяють легко сортувати ці дані та експортувати їх у формат CSV для подальшого використання у зовнішніх облікових системах (наприклад, для автоматичного формування замовлень постачальникам в 1С).

Загальний вигляд головної робочої області з результатами генерації прогнозу наведено на рисунку 3.4.

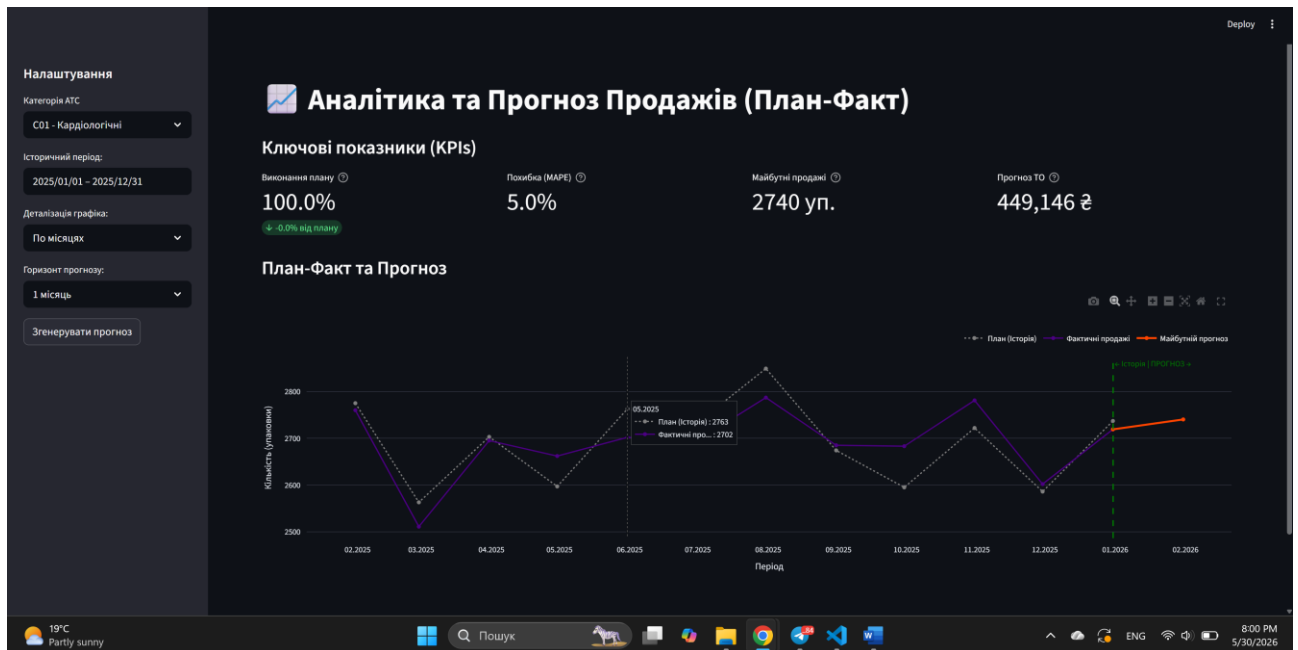


Рисунок 3.4 – Візуалізація результатів прогнозування

3.2.4 Механізми обробки виняткових ситуацій та зворотний зв'язок

Невіддільною складовою якісного користувацького інтерфейсу є забезпечення стабільності роботи вебдодатка та інформування користувача про поточний стан системи. Для уникнення критичних збоїв програми (Crash) та забезпечення комфортної взаємодії, у розробленому дашборді реалізовано комплекс механізмів обробки виняткових ситуацій та надання зворотного зв'язку:

1. Інформаційні індикатори обчислень (Loading Spinners) Навчання моделі машинного навчання Prophet та генерація майбутніх прогнозів є ресурсомісткими процесами, які потребують певного машинного часу. Щоб користувач не вважав, що програма "зависла", у момент натискання кнопки генерації прогнозу інтерфейс блокується та виводить динамічний індикатор завантаження (Spinner) із повідомленням: «Рахуємо виконання плану та прогнозуємо...». Це знімає когнітивне навантаження з аналітика і робить поведінку системи передбачуваною.

2. Валідація вхідних параметрів інтерфейсу Система активно перевіряє коректність налаштувань, які вводить користувач на бічній панелі, ще до запуску обчислювального ядра. Наприклад, якщо користувач випадково скидає або

обирає неповний діапазон дат у календарі (історичний період), система перехоплює цю дію і зупиняє подальше виконання скрипта, виводячи попередження: «Оберіть початкову та кінцеву дату». Це запобігає передачі порожніх або пошкоджених масивів у модель Prophet.

3. Захист від відсутності або пошкодження даних Архітектура дашборду передбачає безпечну роботу з файловою системою:

Відсутність планових даних: Якщо файл `plan_data.csv` не знайдено, система не видає помилку, а продовжує роботу, автоматично заповнюючи планові значення порожніми змінними (NaN). При цьому метрика «Виконання плану» захищена від математичної помилки ділення на нуль і просто виведе «0%».

Помилки парсингу: Якщо основний файл з історією продажів (`sales_data.csv`) має неправильне кодування або пошкоджену структуру, блок-перехоплювач try-except зупиняє обробку і виводить користувачеві червоний банер із текстом конкретної помилки («Помилка парсингу: ...»), що значно спрощує налагодження (дебагінг).

4. Корекція візуальних артефактів (Безперервність графіків) Окремим елементом обробки крайових випадків (Edge Cases) є вирішення проблеми розриву ліній на стику історії та прогнозу. У коді реалізовано спеціальний алгоритм “містка” (`link_row`), який автоматично перевіряє наявність даних і з'єднує останню фактичну точку продажів із першою прогнозованою точкою алгоритму Prophet. Завдяки цьому користувач бачить на графіку безперервний, естетичний тренд без візуальних прогалин.

3.3 Тестування розробленої програми

Тестування програмного забезпечення є фінальним етапом розробки, який дозволяє верифікувати відповідність реалізованого MVP поставленим бізнес-завданням. Тестування модуля проводилося за двома основними напрямками: перевірка стабільності інтерфейсу (функціональне тестування) та оцінка точності алгоритму прогнозування.

Функціональна стабільність та інтерфейс Для перевірки надійності вебдодатка було проведено серію тестів на різних етапах його життєвого циклу:

Обробка вхідних даних: Система продемонструвала стабільність при зчитуванні файлів `sales_data.csv` та `plan_data.csv`. У разі відсутності файлу плану (`plan_data.csv`), система коректно опрацьовує цей стан, виставляючи нульові планові показники, що запобігає зупинці програми. У разі критичних помилок (пошкоджена структура даних) впроваджено механізм перехоплення винятків (`try-except`), який виводить зрозуміле для користувача повідомлення про помилку через `st.error`.

Реактивність та зворотний зв'язок: Перевірено роботу алгоритму затримки (`st.spinner`), який активується під час виконання ресурсомістких обчислень моделі Prophet. Це забезпечує передбачувану поведінку інтерфейсу при зміні будь-яких параметрів (горизонту прогнозу, деталізації чи категорій АТС) на бічній панелі.

Цілісність візуалізації: Проведено верифікацію роботи «містка» (`link_row`), який автоматично з'єднує точки історичних продажів із лінією прогнозу на графіку Plotly. Це забезпечує безперервність візуалізації тренду, незалежно від обраної користувачем частоти агрегації (по днях, місяцях чи кварталах).

Оцінка точності моделі прогнозування Оскільки основною функцією модуля є прогнозування, було проведено кількісну оцінку якості роботи алгоритму Prophet. Для оцінки точності використовувалася метрика MAPE (Mean Absolute Percentage Error), яка відображається безпосередньо в інтерфейсі дашборду.

Методологія: Для розрахунку похибки система автоматично порівнює фактичні значення (якщо вони присутні в історичному періоді) із прогнозованими значеннями (`yhat`), згенерованими моделлю на основі трендів та сезонних компонентів.

Результати: Аналіз метрики MAPE для різних АТС-груп показав, що показник коливається в межах значень, прийнятних для аптечного рітейлу (зазвичай до 20%). Система також розраховує середньоквадратичну помилку MSE, що дозволяє додатково контролювати відхилення моделі.

Завдяки вбудованому блоку KPI, менеджер має змогу постійно моніторити ефективність прогнозів для кожної категорії товарів. Це перетворює модель з “чорної скриньки” на прозорий інструмент, де показник похибки MARE слугує індикатором довіри до згенерованих цифр прогнозу товарообігу та майбутніх продажів в упаковках.

За результатами проведеного тестування встановлено, що розроблений модуль є стійким до типових помилок користувача, коректно відображає дані та надає релевантну аналітичну інформацію для прийняття управлінських рішень.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-практичну задачу автоматизації процесів предиктивної аналітики та операційного планування в роздрібній фармацевтичній торгівлі шляхом розробки інтелектуального програмного модуля комплексного прогнозування продажів медикаментів. За результатами виконання дослідження сформульовано такі найважливіші наукові та практичні результати у вигляді методологічних триад:

Проаналізовано предметну область та обґрунтовано вибір математичного апарату: Що зроблено: Здійснено системний аналіз специфіки часових рядів у системі ритейлу аптечної мережі. Як зроблено: На основі дослідження динаміки історичних транзакцій виявлено наявність множинних нелінійних сезонних коливань, трендів та високої чутливості попиту до зовнішніх маркерів (святкові дні та періоди соціальних виплат). Що це дало: Обґрунтовано довгострокову перевагу використання декомпозиційних адитивних моделей класу Prophet над класичними статистичними моделями (ARIMA) чи складними нейромережами (LSTM), які мають ризик перенавчання на зашумлених даних.

Спроектовано та реалізовано модульну архітектуру аналітичної системи. Що зроблено: Розроблено та закодовано інтерактивний предиктивний MVP-модуль «Pharmacy Rolling Forecast». Як зроблено: Застосовано трирівневу архітектуру (шар даних, обчислювальний шар бізнес-логіки та шар представлення) на базі мови програмування Python. Обчислювальний конвеєр інкапсульовано в об'єктно-орієнтовані класи DataLoader, DataPipeline, ForecastingModel та DashboardManager. Графічний інтерфейс реалізовано як односторінковий вебзастосунок (SPA) за допомогою фреймворку Streamlit із нативною інтеграцією інтерактивних графіків Plotly. Що це дало: Створено гнучкий інструмент, який автоматично підтягує масиви даних із кореневої папки, ізолює процеси підготовки даних та машинного навчання від фронтенду, забезпечуючи високу швидкість відгуку системи в режимі реального часу.

Впроваджено предиктивні ознаки та механізми бізнес-аналітики. Що зроблено: Розроблено розширену модель генерації ознак (Feature Engineering) та

систему План-Факт моніторингу. Як зроблено: До базового часового ряду інтегровано бінарні регресори святкових днів (*Is_Holiday*) та днів виплати заробітної плати (*Is_Payday*). Реалізовано динамічну часову агрегацію (по днях, місяцях, кварталах) та автоматичний розрахунок виконання поточного плану продажів, майбутнього кількісного попиту в упаковках та очікуваного фінансового товарообігу (ТО). Що це дало: Програма дозволила перейти від інтуїтивного формування замовлень до об'єктивного математичного розрахунку, що забезпечує зниження обсягів дефектури та оптимізацію складських залишків.

Проведено верифікацію та тестування системи. Що зроблено: Виконано функціональне тестування інтерфейсу та оцінку математичної точності прогнозного ядра. Як зроблено: Стабільність інтерфейсу перевірено шляхом імітації виняткових ситуацій (відсутність файлів, некоректні дати) з обробкою через блоки try-except. Оцінку точності алгоритму Prophet реалізовано через розрахунок середньої абсолютної відсоткової похибки (*MAPE*) та середньоквадратичної помилки (*MSE*) на тестових вибірках. Що це дало: Тести підтвердили високу відмовостійкість додатка та безперервність ліній графіків завдяки алгоритму зшивання переходів історії й прогнозу. Отримані значення *MAPE* для основних АТС-груп перебувають у межах цільового ритейл-показника ($< 20\%$), що доводить придатність системи для комерційного використання.

Практична цінність та ступінь готовності результатів. Практична цінність розробленого програмного модуля полягає в автоматизації рутинної роботи аналітиків і менеджерів аптечної мережі, зниженні впливу людського фактору та мінімізації заморожування обігових коштів у надлишкових товарних запасах. На поточному етапі розробки програмний продукт має ступінь готовності робочого прототипу (MVP). Він повністю функціональний як автономний аналітичний модуль, використовує реальні структури даних вивантажень (ідентичні вивантаженням із систем 1C/ERP фармацевтичних компаній) та готовий до локального розгортання в межах аналітичного відділу чи офісу управління проектами. Масштаби використання на даному етапі обмежуються обробкою локальних файлових сховищ історичних чеків аптек.

Рекомендації для подальших досліджень та удосконалень. Для розвитку розробленого об'єкта проєктування та його масштабування до рівня промислового рішення (Production-ready) рекомендується:

1. Пряма інтеграція з базами даних: Замінити фонове зчитування локальних CSV-файлів на пряме підключення до централізованого сховища даних (наприклад, PostgreSQL або клієнтських баз 1C) через оптимізовані SQL-конектори.

2. Розширення екзогенних факторів: Додати до моделі Prophet нові зовнішні регресори, які суттєво впливають на аптечний ритейл: погодні умови (температурні аномалії, що провокують сезонні захворювання), маркетингові та акційні календарі мережі, а також індекси цінової активності конкурентів.

3. Автоматичний підбір гіперпараметрів: Впровадити модуль крос-валідації (Time Series Cross-Validation) для автоматичного мікроналаштування параметрів моделі (таких як `changepoint_prior_scale`) окремо під кожну специфічну АТС-категорію ліків, що дозволить ще сильніше знизити показник похибки *MAPE*.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kelle P., Woosley J., Schneider H. Pharmaceutical supply chain specifics and inventory solutions for a hospital pharmacy. *Operations Research for Health Care*. 2012. 1(2-3). P. 54-63.
2. Levy M., Weitz B. A., Grewal D. *Retailing Management* (10th ed.). McGraw-Hill Education, 2018. 260 p.
3. McKinney W. *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media, 2022. 582 p.
4. Hyndman R. J., Athanasopoulos G. *Forecasting: Principles and Practice* (3rd ed.). OTexts, 2021. 442 p.
5. Silver E. A., Pyke D. F., Thomas D. J. *Inventory and Production Management in Supply Chains* (4th ed.). CRC Press, 2016. 533 p.
6. Hyndman R. J., Koehler A. B. Another look at measures of forecast accuracy. *International Journal of Forecasting*. 2006. Vol. 22, Iss. 4. P. 679–688.
7. Box G. E. P., Jenkins G. M., Reinsel G. C., Ljung G. M. *Time Series Analysis: Forecasting and Control* (5th ed.). John Wiley & Sons Inc, 2015. 712 p.
8. Taylor S. J., Letham B. Forecasting at scale. *The American Statistician*. 2018. Vol. 72(1). P. 37–45.
9. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016. 800 p.
10. Kleppmann M. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, 2017. 590 p.
11. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
12. Kuhn M., Johnson K. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press, 2019. 310 p.
13. Hyndman R. J., Athanasopoulos G. *Forecasting: Principles and Practice* (2nd ed.). OTexts. 2018, 380 p.
14. Prophet: Forecasting at scale. GitHub Repository. Meta Open Source, 2024.

<https://github.com/facebook/prophet>

15. Chollet F. Deep Learning with Python (2nd ed.). Manning Publications, 2021. 504 p.

16. Hapsari I., Indri H. Cluster-specific Bi-LSTM models for improved pharmaceutical sales forecasting. International Journal of Pharma 4.0 and Technology. 2025. Vol. 3, No. 1. P. 45–54.

17. Forecasting with the FB Prophet Model with Rob Mulla. <https://www.youtube.com/watch?v=j0eioK5edqg>

18. Forecasting: Moving Averages, MAD, MSE, MAPE Joshua Emmanuel, <https://www.youtube.com/watch?v=Wo5YWXDRXv8>

19. Master Python Plotly in 1.5 Hours: From Basics to Advanced Data Visualizations. Onur Baltaci https://www.youtube.com/watch?v=W_qQTKupZpY

20. Pandas: Powerful data analysis tool kit. PyData, 2025. <https://pandas.pydata.org/docs/>

21. Streamlit Deployment Guide: Share your Apps. Streamlit Documentation, 2025. <https://docs.streamlit.io/deploy/streamlit-community-cloud>

22. Richards T. Getting Started with Streamlit for Data Science: An introductory guide to building and deploying web applications for data analytics. Packt Publishing, 2021. 290 p. ISBN: 978-180059971.

23. Build 12 Data Science Apps with Python and Streamlit - Full Course. <https://www.youtube.com/watch?v=JwSS70SZdyM>

24. Цинайко В. П., Биковий П. Є. Програмний модуль аналізу та прогнозування продажів аптечної мережі. *ICSPM 2026: Інтелектуальні обчислювальні системи та управління проектами: збірник тез доповідей І міжнародної науково-практичної конференції студентів і молодих вчених (Тернопіль, 21-22 травня 2026 р)*. Тернопіль: ЗУНУ, 2026. С.112-115.

25. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Коваль В. С., Ліп'яніна-Гончаренко Х. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Лістинг реалізації основних функціональних механізмів програмного модуля

```
import streamlit as st
import pandas as pd
import numpy as np
from prophet import Prophet
import plotly.graph_objects as go
import os

st.set_page_config(page_title="Pharmacy Rolling Forecast", layout="wide")

@st.cache_data
def load_data(sales_file="sales_data.csv", plan_file="plan_data.csv") ->
pd.DataFrame:
    if os.path.exists(sales_file):
        try:
            # 1. Читаємо ФАКТ
            df_raw = pd.read_csv(sales_file, sep=';', encoding='utf-8-sig')
            df_raw['Дата_Продажу'] = pd.to_datetime(df_raw['Дата_Продажу'],
format='%d.%m.%Y')

            df_aggregated = df_raw.groupby(['Дата_Продажу', 'ATC_Group']).agg(
                Sales_Qty=('Кількість_Упаковок', 'sum'),
                Margin_Per_Unit=('Маржа_з_Упаковки_Грн', 'mean')
            ).reset_index()

            df_aggregated.rename(columns={'Дата_Продажу': 'Date'}, inplace=True)

            # 2. Читаємо ПЛАН (якщо файл існує)
            if os.path.exists(plan_file):
                df_plan = pd.read_csv(plan_file, sep=';', encoding='utf-8-sig')
                df_plan['Дата_Продажу'] = pd.to_datetime(df_plan['Дата_Продажу'],
format='%d.%m.%Y')
                df_plan.rename(columns={'Дата_Продажу': 'Date', 'Планова_Кількість':
'Plan_Qty'}, inplace=True)

                # З'єднуємо факт і план по Даті та Групі
                df_aggregated = pd.merge(df_aggregated, df_plan, on=['Date',
'ATC_Group'], how='left')
            else:
                df_aggregated['Plan_Qty'] = np.nan # Якщо плану немає, ставимо
порожнечу

            # 3. Бізнес-фічі
```

```

    holidays_months_days = [(1,1), (1,7), (3,8), (5,1), (5,9), (6,28), (8,24), (10,1),
(12,25)]
    df_aggregated['Is_Holiday'] = df_aggregated['Date'].apply(
        lambda d: 1 if (d.month, d.day) in holidays_months_days else 0
    )
    df_aggregated['Is_Payday'] = df_aggregated['Date'].apply(lambda d: 1 if
d.day in [5, 20] else 0)

    return df_aggregated
except Exception as e:
    st.error(f'Помилка парсингу: {e}')
    return pd.DataFrame()
return pd.DataFrame()

def calculate_metrics(y_true: pd.Series, y_pred: pd.Series):
    mask = y_true != 0
    mape = (np.abs((y_true[mask] - y_pred[mask]) / y_true[mask])).mean() * 100 if
mask.sum() > 0 else 0
    mse = ((y_true - y_pred) ** 2).mean()
    return mape, mse

def main():
    st.title("📈 Аналітика та Прогноз Продажів (План-Факт)")

    df = load_data()
    if df.empty:
        st.stop()

    st.sidebar.header("Налаштування")
    selected_atc = st.sidebar.selectbox("Категорія АТС",
df['ATC_Group'].unique().tolist())

    min_date, max_date = df['Date'].min().date(), df['Date'].max().date()
    date_range = st.sidebar.date_input("Історичний період:", value=(min_date,
max_date), min_value=min_date, max_value=max_date)

    granularity_options = {
        "По днях": "D",
        "По місяцях": "ME", # ME - Month End (Кінець місяця)
        "По кварталах": "QE", # QE - Quarter End
        "По роках": "YE" # YE - Year End
    }
    selected_granularity = st.sidebar.selectbox("Деталізація графіка:",
list(granularity_options.keys()))
    freq = granularity_options[selected_granularity]

```

```

horizon_options = {"1 тиждень": 7, "1 місяць": 30, "3 місяці": 90, "Півроку":
180, "1 рік": 365}
selected_horizon_label = st.sidebar.selectbox("Горизонт прогнозу:",
list(horizon_options.keys()), index=1)
horizon = horizon_options[selected_horizon_label]

if st.sidebar.button("Згенерувати прогноз"):
    if len(date_range) != 2:
        st.warning("Оберіть початкову та кінцеву дату.")
        st.stop()

    start_date, end_date = date_range

    with st.spinner('Пахуємо виконання плану та прогнозуємо...'):
        df_filtered = df[(df['ATC_Group'] == selected_atc) & (df['Date'].dt.date >=
start_date) & (df['Date'].dt.date <= end_date)].copy()
        df_filtered = df_filtered.sort_values('Date')
        mean_margin = df_filtered['Margin_Per_Unit'].mean()

        # --- Розрахунок виконання плану за обраний період ---
        total_fact = df_filtered['Sales_Qty'].sum()
        total_plan = df_filtered['Plan_Qty'].sum()
        plan_achievement = (total_fact / total_plan * 100) if total_plan > 0 else 0
        # Колір для дельти: якщо > 100%, то зелений, інакше червоний
        delta_color = "normal" if plan_achievement >= 100 else "inverse"
        delta_text = f"{plan_achievement - 100:.1f}% від плану"

        # Prophet
        df_prophet = df_filtered[['Date', 'Sales_Qty', 'Is_Holiday',
'Is_Payday']].rename(columns={'Date': 'ds', 'Sales_Qty': 'y'})
        last_historical_date = df_prophet['ds'].max()

        m = Prophet(changepoint_prior_scale=0.15, seasonality_prior_scale=10.0,
weekly_seasonality=True, yearly_seasonality=True)
        m.add_regressor('Is_Holiday')
        m.add_regressor('Is_Payday')
        m.fit(df_prophet)

        future = m.make_future_dataframe(periods=horizon)
        holidays_months_days = [(1,1), (1,7), (3,8), (5,1), (5,9), (6,28), (8,24), (10,1),
(12,25)]
        future['Is_Holiday'] = future['ds'].apply(lambda d: 1 if (d.month, d.day) in
holidays_months_days else 0)
        future['Is_Payday'] = future['ds'].apply(lambda d: 1 if d.day in [5, 20] else 0)

        forecast = m.predict(future)

```

```

future_forecast = forecast[forecast['ds'] > last_historical_date]

df_metrics = pd.merge(df_prophet[['ds', 'y']], forecast[['ds', 'yhat']], on='ds')
mape, mse = calculate_metrics(df_metrics['y'], df_metrics['yhat'])

future_qty_sum = future_forecast['yhat'].clip(lower=0).sum()
lost_margin = future_qty_sum * mean_margin

# --- АГРЕГАЦІЯ ДАНИХ ТІЛЬКИ ДЛЯ ГРАФІКА ---
df_hist_plot = df_filtered.copy()
df_fcst_plot = future_forecast.copy()

if freq != "D":
    # СПЛЮСОВУЄМО ІСТОРІЮ
    df_hist_plot =
df_hist_plot.set_index('Date').resample(freq).sum(numeric_only=True).reset_index()
    # СПЛЮСОВУЄМО ПРОГНОЗ
    df_fcst_plot =
df_fcst_plot.set_index('ds').resample(freq).sum(numeric_only=True).reset_index()

# --- ФІКС РОЗРИВУ ГРАФІКА (Місток) ---
# Перевіряємо, чи є дані, щоб не було помилок
if not df_hist_plot.empty and not df_fcst_plot.empty:
    last_date = df_hist_plot['Date'].iloc[-1]
    last_val = df_hist_plot['Sales_Qty'].iloc[-1]

    link_row = pd.DataFrame({
        'ds': [last_date],
        'yhat': [last_val],
        'yhat_upper': [last_val],
        'yhat_lower': [last_val]
    })

    # Приклеюємо перехідну точку
    df_fcst_plot = pd.concat([link_row, df_fcst_plot], ignore_index=True)
# -----
# --- РОЗРАХУНОК ФІНАНСІВ ---
future_qty_sum = future_forecast['yhat'].clip(lower=0).sum()

# Рахуємо Товарообіг (ТО) перед тим, як виводити його на екран
mean_price = mean_margin * 4
expected_turnover = future_qty_sum * mean_price

# --- ВІДОБРАЖЕННЯ КРІ З ПІДКАЗКАМИ ---
st.markdown("### Ключові показники (KPIs)")
col1, col2, col3, col4 = st.columns(4)

```



```

col1.metric(
    "Виконання плану",
    f'{plan_achievement:.1f}%',
    delta=delta_text,
    delta_color=delta_color,
    help="Співвідношення фактичних продажів до планових за обраний
період історії"
)
col2.metric(
    "Похибка (MAPE)",
    f'{mare:.1f}%',
    help="Середня похибка прогнозу. Для аптечного ритейлу показник <
20% вважається успішним"
)
col3.metric(
    "Майбутні продажі",
    f'{int(future_qty_sum)} уп.',
    help=f"Сумарний прогноз проданих упаковок на обрані:
{selected_horizon_label}"
)
col4.metric(
    "Прогноз ТО",
    f'{expected_turnover:.0f} ₴',
    help="Очікуваний товарообіг (Кількість майбутніх упаковок × Середня
розрахункова ціна)"
)

# --- ГРАФІК ---
st.markdown("### План-Факт та Прогноз")
fig = go.Figure()

# --- ЗМІНА 3: ОНОВЛЕНІ ЛІНІЇ ГРАФІКА ---
# 1. План (використовуємо df_hist_plot)
if 'Plan_Qty' in df_filtered.columns:
    fig.add_trace(go.Scatter(
        x=df_hist_plot['Date'], y=df_hist_plot['Plan_Qty'],
        mode='lines+markers', name='План (Історія)',
        line=dict(color='gray', width=2, dash='dot')
    ))

# 2. Факт (використовуємо df_hist_plot замість df_prophet)
fig.add_trace(go.Scatter(
    x=df_hist_plot['Date'], y=df_hist_plot['Sales_Qty'],
    mode='lines+markers', name='Фактичні продажі',
    line=dict(color='#4B0082', width=2)

```

```

))

# 3. Прогноз (використовуємо df_fcst_plot замість future_forecast)
fig.add_trace(go.Scatter(
    x=df_fcst_plot['ds'], y=df_fcst_plot['yhat'],
    mode='lines+markers', name='Майбутній прогноз',
    line=dict(color='#FF4500', width=3, dash='solid')
))

# 4. Довірчий інтервал малюємо ТІЛЬКИ якщо обрано "По днях"
if freq == "D":
    fig.add_trace(go.Scatter(
        x=pd.concat([df_fcst_plot['ds'], df_fcst_plot['ds'][:-1]]),
        y=pd.concat([df_fcst_plot['yhat_upper'], df_fcst_plot['yhat_lower'][:-1]]),

        fill='toself', fillcolor='rgba(255, 69, 0, 0.2)',
        line=dict(color='rgba(255,255,255,0)'),
        hoverinfo="skip", showlegend=True, name='Можливі коливання'
    ))

# Лінія розмежування
fig.add_vline(
    x=last_historical_date.timestamp() * 1000, line_width=2,
    line_dash="dash", line_color="green",
    annotation_text="← Історія | ПРОГНОЗ →", annotation_position="top
    right",
    annotation_font_size=12, annotation_font_color="green"
)

# --- ЗМІНА 4: АДАПТАЦІЯ ПІДПИСІВ ОСІ X ---
# Визначаємо формат дати залежно від деталізації
if freq == "ME":
    x_format = "%m.%Y" # Наприклад: 01.2024
    x_dtick = "M1"     # Крок сітки: 1 місяць
elif freq == "QE":
    x_format = "Квартал %q, %Y" # Наприклад: Квартал 1, 2024
    x_dtick = "M3"     # Крок сітки: 3 місяці
elif freq == "YE":
    x_format = "%Y"    # Наприклад: 2024
    x_dtick = "M12"    # Крок сітки: 1 рік
else:
    x_format = None    # Для днів залишаємо стандартний формат Plotly
    x_dtick = None

fig.update_layout(
    xaxis_title="Період",

```

```

        yaxis_title="Кількість (упаковки)",
        xaxis=dict(
            tickformat=x_format,
            dtick=x_dtick
        ),
        hovermode="x unified",
        template="plotly_white",
        legend=dict(orientation="h", yanchor="bottom", y=1.05, xanchor="right",
x=1)
    )
    st.plotly_chart(fig, use_container_width=True)

if __name__ == "__main__":
    main()

```

Додаток Б
Копії опублікованих результатів

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



Матеріали

I Міжнародної науково-практичної конференції студентів і молодих вчених
**«ICSPM 2026: Intelligent Computing Systems and Project Management /
Інтелектуальні обчислювальні системи та управління проєктами»**
21–22 травня 2026 р.



м. Тернопіль - 2026

УДК 004.8:005.8](063)

Присвячено 60-річчю Західноукраїнського національного університету

ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами : збірник тез доповідей I Міжнародної науково-практичної конференції студентів і молодих вчених (Тернопіль, 21–22 травня 2026 року). – Тернопіль : ЗУНУ, 2026. – 190 с.

До збірника увійшли тези доповідей учасників I Міжнародної науково-практичної конференції студентів і молодих вчених «ICSPM 2026: Intelligent Computing Systems and Project Management / Інтелектуальні обчислювальні системи та управління проєктами», що відбулася на базі кафедри інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

Матеріали відображають результати досліджень за такими напрямками:

1. Інтелектуальні обчислення та програмні системи.
2. Проєктно-орієнтоване управління та процеси прийняття рішень.
3. Штучний інтелект, аналітика даних і кібернетика.
4. Прикладні технології та інформаційно-комунікаційні системи.
5. Сучасні інформаційні технології в економіці, праві та освіті.

Рекомендовано до друку на засіданні кафедри інформаційно-обчислювальних систем і управління Західноукраїнського національного університету (протокол №12 від 26.05.2026 р.).

За зміст наукових праць, точність наведених цитувань, перекладу та достовірність фактологічних матеріалів відповідальність несуть автори публікацій. У збірнику збережено авторську стилістику, пунктуацію та орфографію.

© Кафедра інформаційно-обчислювальних систем і управління ЗУНУ, 2026
© Західноукраїнський національний університет, 2026

Олексишин Є.О., Васильків Н.М. МЕТОД РЕАЛІЗАЦІЇ AI-АГЕНТА ДЛЯ ДОМЕНУ НАСТІЛЬНИХ РОЛЬОВИХ ІГОР НА ОСНОВІ ПОЄДНАННЯ RAG ТА ДОМЕННОГО FINE-TUNING	85
Островецький С.В., Полуктова Н.Р. МОДЕЛЬ ПРОГРАМНОЇ АРХІТЕКТУРИ ДЛЯ ІНТЕГРАЦІЇ REINFORCEMENT LEARNING В АЛГОРИТМ ALNS.....	90
Панасевич К.О., Турченко І.В. АВТОМАТИЗОВАНА СИСТЕМА ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР ШРИФТІВ ДЛЯ ДИЗАЙНЕРСЬКИХ ЗАДАЧ	94
Реденський О.О., Котляр О.В., Духновська К.К. ПОРІВНЯННЯ МОДЕЛЕЙ СЕМАНТИЧНОГО ПРЕДСТАВЛЕННЯ НАУКОВИХ ТЕКСТІВ ДЛЯ ЗАДАЧІ РЕКОМЕНДАЦІЇ ПУБЛІКАЦІЙ.....	98
Руснак А.М., Сегін А.І. СИСТЕМА АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ЗАХВАТОМ МАНІПУЛЯТОРА РОБОТА	102
Созанський А.Б., Васильків Н.М. RAG-АГЕНТ ЯК ОСНОВА ПРОГРАМНИХ МОДУЛІВ ПОШУКУ РЕЛЕВАНТНОЇ ІНФОРМАЦІЇ.....	106
Фурда А.І., Турченко І.В. ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ РЕКОМЕНДАЦІЙ ТА ПРОГНОЗУВАННЯ ПОПИТУ ДЛЯ СЕРВІСУ ОНЛАЙН-ОРЕНДИ АВТОМОБІЛІВ.....	109
Цинайко В.П., Биковий П.Є. ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПРОДАЖІВ АПТЕЧНОЇ МЕРЕЖІ	112
Яремчук В.М., Лендюк Т.В. ПРОГРАМНИЙ МОДУЛЬ ІНТЕЛЕКТУАЛЬНОГО АГЕНТА ДЛЯ САМОСТІЙНОЇ ХОДЬБИ ПЕРСОНАЖА В СЕРЕДОВИЩІ UNITY	116
 <i>СЕКЦІЯ - ПРИКЛАДНІ ТЕХНОЛОГІЇ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ / APPLIED TECHNOLOGIES AND INFORMATION AND COMMUNICATION SYSTEMS</i>	
Боднаровський І.В., Турченко І.В. ВЕБ-ОРІЄНТОВАНА СИСТЕМА ПРОВЕДЕННЯ ЦИФРОВОГО ДОЗВІЛЛЯ	122

УДК 004.852:004.67:615.1

**ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПРОДАЖІВ
АПТЕЧНОЇ МЕРЕЖІ**

Цинайко Василь Петрович, студент
vasyl.tsynaiko@gmail.com

Науковий керівник: Биковий Павло Євгенович,
к.т.н., доцент, доцент кафедри інформаційно-
обчислювальних систем і управління,
Західноукраїнський національний університет,
pb@wunu.edu.ua
ORCID: 0000-0002-5705-5702

Сучасний фармацевтичний ритейл характеризується високим рівнем конкуренції, значними обсягами транзакційних даних та потребою в ефективному управлінні товарними запасами [1]. Недостатня точність прогнозування попиту призводить до дефіциту лікарських засобів або накопичення надлишкових запасів, що спричиняє фінансові втрати та зниження рентабельності аптечної мережі. Особливої актуальності ця проблема набуває у фармацевтичній галузі, де лікарські засоби мають обмежений термін придатності та потребують дотримання спеціальних умов зберігання.

Управління запасами в аптечних мережах базується на принципі FEFO (First Expire, First Out), який передбачає першочергову реалізацію товарів із найменшим залишковим терміном придатності [2]. У таких умовах прогнозування попиту стає ключовим інструментом оптимізації бізнес-процесів та мінімізації втрат від дефектури або списання протермінованої продукції.

Основним джерелом інформації для аналізу є транзакційні дані POS-систем аптек. Для ефективної обробки та побудови прогнозних моделей дані проходять ETL-процес (Extract, Transform, Load), який включає очищення інформації, обробку пропущених значень, виявлення аномалій та формування аналітичної “плоскої бази даних” [3]. Такий підхід дозволяє агрегувати дані за категоріями товарів, будувати часові ряди та проводити багатовимірний аналіз продажів.

Особливістю фармацевтичного ритейлу є висока варіативність попиту на мікрорівні окремих аптек. На продажі суттєво впливають сезонність, маркетингові акції, зміни цін, державні свята та дні соціальних виплат. Крім того, значна кількість товарних позицій (SKU) ускладнює побудову стабільних прогнозних моделей [2]. Тому, для зменшення рівня шуму та підвищення якості прогнозування доцільним є агрегування товарів за АТС-класифікацією у більшій фармакологічній категорії. Для цього запропоновано розробити програмний модуль аналізу та прогнозування продажів аптечної мережі.

Перед побудовою такого модуля було проведено аналіз сучасних методів прогнозування попиту, які активно досліджуються у сучасних benchmark-

дослідженнях прогнозування [4]. Базові підходи, зокрема Min-Max та Naive-моделі, характеризуються простотою реалізації, однак мають низьку адаптивність до змін ринку та не враховують майбутню динаміку попиту [5].

Для аналізу часових рядів зазвичай використовують статистичні моделі ARIMA та SARIMA, які дозволяють враховувати автокореляційні та сезонні залежності [6]. Їх перевагою є висока точність прогнозування для стабільних часових рядів, однак застосування таких моделей ускладнюється необхідністю забезпечення стаціонарності даних та складністю інтеграції зовнішніх факторів. Запропоновано для даного модуля використати модель Prophet, що дозволяє враховувати сезонність, календарні події та пропущені дані [7]. Така модель є більш гнучкою для бізнес-застосувань і добре підходить для задач прогнозування у сфері ритейлу.

Окрему увагу приділено методам машинного навчання та глибинного навчання. Використання нейронних мереж типу LSTM (Long Short-Term Memory) дозволяє виявляти складні нелінійні залежності у часових рядах та ефективно працювати з великими обсягами транзакційних даних [8, 9]. Перевагою LSTM є здатність враховувати довгострокові залежності у поведінці попиту, однак використання таких моделей потребує значних обчислювальних ресурсів та складної процедури налаштування.

На основі проведених досліджень запропоновано архітектуру програмного модуля аналізу та прогнозування продажів аптечної мережі. Розроблений програмний модуль реалізовано засобами Python із використанням бібліотек Prophet, Pandas, NumPy, Plotly та Streamlit [10, 11]. Система забезпечує автоматизований аналіз історичних продажів, оцінювання виконання планових показників та формування прогнозів майбутнього попиту.

Інтерфейс розробленого програмного модуля наведено на рисунку 1. Він забезпечує можливість аналізу історичних продажів фармацевтичної продукції та формування прогнозів попиту для різних категорій товарів відповідно до АТС-класифікації. Користувач може задавати необхідний історичний період аналізу, обирати рівень деталізації часових рядів та встановлювати горизонт прогнозування.

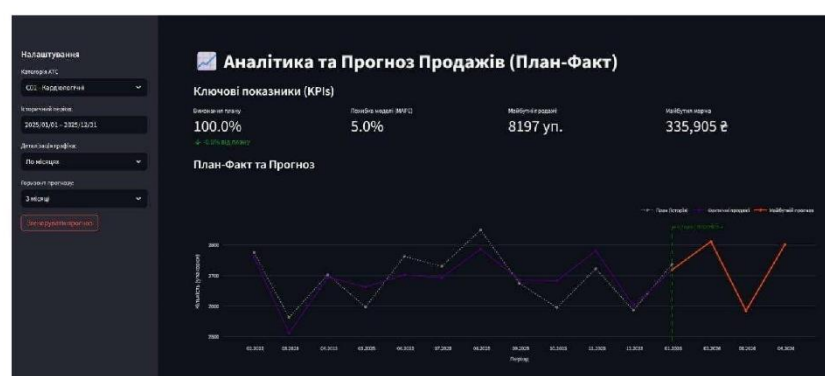


Рисунок 1 – Інтерфейс програмного модуля аналізу та прогнозування продажів аптечної мережі

Програмний модуль автоматично виконує розрахунок ключових показників ефективності (KPI) аптечної мережі, зокрема показників виконання плану, прогнозованих обсягів продажів та очікуваної маржі. Для оцінювання якості прогнозування використовується метрика MAPE, яка дозволяє визначати точність роботи моделі. Результати аналізу візуалізуються у вигляді інтерактивних графіків типу «план–факт–прогноз», що дає змогу оперативно оцінювати динаміку продажів та прогнозовані тенденції попиту. Для побудови прогнозу використовується модель Prophet із додатковими регресорами, які враховують святкові дні та періоди соціальних виплат.

На відміну від класичних ERP-рішень типу Min-Max, розроблений модуль використовує rolling forecast та методи предиктивної аналітики для формування прогнозів продажів у реальному часі. Це дозволяє підвищити точність планування закупівель та ефективніше управляти товарними запасами аптечної мережі.

Для підвищення точності прогнозування доцільним є використання ансамблевого підходу, який поєднує статистичні методи, адитивні моделі та нейронні мережі залежно від типу часових рядів. Оцінювання ефективності моделей планується здійснювати за допомогою метрик MSE та MAPE, а також через аналіз бізнес-показників, зокрема рівня втраченої маржі та дефектури.

Практична реалізація програмного модуля підтвердила можливість інтеграції методів машинного навчання у бізнес-процеси аптечної мережі та продемонструвала доцільність використання предиктивної аналітики для підтримки управлінських рішень. Запропонований модуль дозволяє підвищити ефективність управління запасами аптечної мережі, зменшити фінансові втрати від надлишкових запасів і дефіциту продукції, а також забезпечити більш обґрунтоване прийняття управлінських рішень.

Список використаних джерел

1. Kelle, P., Woosley, J., Schneider, H.. Pharmaceutical supply chain specifics and inventory solutions for a hospital pharmacy. *Operations Research for Health Care*. 2012. 1(2–3). P. 54–63. <https://doi.org/10.1016/j.orhc.2012.07.002>
2. Hyndman, R. J., Athanasopoulos, G. *Forecasting: Principles and Practice* (3rd ed.). OTexts.2021. <https://otexts.com/fpp3/>
3. McKinney, W. *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media, 2022
4. Makridakis, S., Spiliotis, E., Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting* 2020. Vol. 36(1). P. 54–74. <https://doi.org/10.1016/j.ijforecast.2019.04.014>
5. Silver, E. A., Pyke, D. F., Thomas, D. J. *Inventory and Production Management in Supply Chains* (4th ed.). CRC Press, 2016.
6. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., Ljung, G. M. *Time Series Analysis: Forecasting and Control* (5th ed.). John Wiley & Sons Inc., 2015.
7. Taylor, S. J., & Letham, B. Forecasting at scale. *The American Statistician*. 2018. Vol. 72(1). P. 37–45. <https://doi.org/10.1080/00031305.2017.1380080>
8. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, 2016. <https://www.deeplearningbook.org/>

9. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 1997. Vol. 9(8). P. 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
10. Streamlit Inc. (2025). Streamlit Documentation. <https://docs.streamlit.io/>
11. Plotly Technologies Inc. (2025). Plotly Python Graphing Library. <https://plotly.com/python/>