

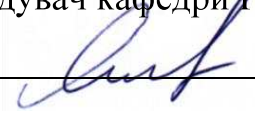
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

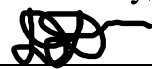
" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

**на тему: Прогнозування попиту та виявлення аномалій із використанням
ML у середовищі .NET**

Виконав: студент групи 6145 м



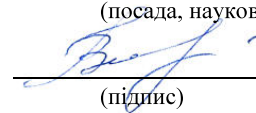
(підпис)

Єфімов Д.Д.

Керівник роботи:

доцент кафедри ІУСТ, PhD.

(посада, науковий ступінь вчене звання)



(підпис)

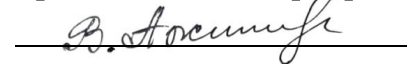
Ворона М.В.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми



к.т.н, доц. Ажищев В.Ф.

"___" _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Єфімову Денису Денисовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET

Керівник роботи PhD Ворона М.В.

Затверджені наказом ректора №1355-уч від "14" листопада 2025 року.

2. Термін подання роботи: 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

Вступ. 1. Аналіз предметної області, розробка концепції на основі прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET. 2. Розробка проектних рішень прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET. 3. Реалізація прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET. 4. Розрахунок економічної ефективності впровадження прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET. 5. Охорона праці та техніка безпеки. 6. Охорона навколишнього середовища. Висновки. Додатки.

5. Перелік презентаційних матеріалів Діаграма варіантів використання; Архітектура системи; Алгоритм роботи «послідовність викликів»; Діаграма послідовності; Діаграма розгортання системи; Фізична модель бази даних прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET; Апробація дослідження; Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ворона М.В. викладач	05.09.2024	22.12.2024
2	Ворона М.В. викладач	23.12.2024	15.06.2025
3	Ворона М.В. викладач	15.06.2025	25.09.2025
4	Ворона М.В. викладач	25.09.2025	14.11.2025
5	Ворона М.В. викладач	25.09.2025	14.11.2025
6	Ворона М.В. викладач	25.09.2025	14.11.2025

7. Дата видачі завдання 05 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

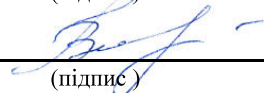
№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2024	
2	Розробка концепції	15.03.2025	
3	Розділ з розробки проектних рішень по системі (та її частинах)	15.06.2025	
4	Реалізація проекту та розробка робочої документації	25.09.2025	
5	Обчислення економічної ефективності	14.11.2025	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2025	
7	Оформлення пояснювальної записки	14.11.2025	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2025	
9	Подання магістерської роботи рецензенту	20.11.2025	
10	Подання магістерської роботи на державний захист	15.12.2025	

Студент


(підпис)

Єфімов Д.Д.
(ПІБ)

Керівник роботи


(підпис)

Ворона М.В.
(ПІБ)

Анотація

У роботі розглядаються методи та інструменти інтелектуального аналізу даних для підтримки процесів управління діяльністю підприємства. Проаналізовано сучасні підходи до збору, обробки та аналізу великих обсягів інформації, що формуються в процесі функціонування підприємств. Обґрунтовано доцільність застосування технологій інтелектуального аналізу даних для виявлення закономірностей, прогнозування ключових показників та підвищення ефективності прийняття управлінських рішень. Розроблено модель, що дозволяє автоматизувати аналітичні процеси та забезпечити керівництво релевантною інформацією для стратегічного планування. Практичне значення роботи полягає у можливості впровадження розроблених підходів у системи підтримки прийняття рішень на підприємствах різних галузей.

Ключові слова: інтелектуальний аналіз даних, управління підприємством, прогнозування, аналітичні моделі, прийняття рішень, ефективність управління.

Abstract

The paper examines methods and tools of data mining to support the processes of enterprise management. Modern approaches to collecting, processing and analyzing large amounts of information generated in the process of enterprise operation are analyzed. The feasibility of using data mining technologies to identify patterns, predict key indicators and increase the efficiency of management decision-making is substantiated. A model has been developed that allows you to automate analytical processes and provide management with relevant information for strategic planning. The practical significance of the paper lies in the possibility of implementing the developed approaches in decision-making support systems at enterprises in various industries.

Keywords: data mining, enterprise management, forecasting, analytical models, decision-making, management efficiency.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. Прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET.....	8
1.1 Аналіз предметної області	8
1.2 Аналіз існуючих недоліків	9
1.3 Варіанти вирішення існуючих недоліків та їх порівняльний аналіз	10
1.4. Концепція системи.....	13
1.5 Математичне забезпечення розробки	15
1.6 Постановка задачі	17
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІС ПРОГНОЗУВАННЯ ПОПИТУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ ІЗ ВИКОРИСТАННЯМ ML У СЕРЕДОВИЩІ .NET В АУТСОРСИНГОВІЙ КОМПАНІЇ, ЩО СПЕЦІАЛІЗУЄТЬСЯ НА РОЗРОБЦІ FULL STACK .NET РІШЕНЬ	21
2.1 Загальна архітектура системи прогнозування	21
2.2 Загально системні рішення	21
2.3 Налаштування пайплайну прогнозування.....	22
2.3 Інтеграція в .NET-додаток	23
2.4 Механізм автоматизації та оновлення моделі	24
2.5 Очікувані результати впровадження	24
2. 6 Проєктування бази даних розробки	26
Висновки до розділу 2	29
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ Прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET	31
3.1 Діаграма розгортання системи	31
3.2 Програмне забезпечення розробки	32
3.3 Вимоги до технічного оснащення розробки	35
Висновки до розділу 3	38

РОЗДІЛ 4. Техніко-економічне обґрунтування розробки системи прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET.....	40
Висновки до розділу 4	41
РОЗДІЛ 5. ОХОРОНА ПРАЦІ	42
5.1 Загальні положення	42
5.2 Характеристика умов праці	42
5.3 Вимоги до робочого місця	43
5.4 Електробезпека.....	43
5.6 Психофізіологічна безпека.....	43
Висновки до розділу 5	44
РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	45
6.1 Загальні положення	45
6.2 Екологічні аспекти розробки	45
6.3 Утилізація електронного обладнання	46
6.4 Використання енергоефективних рішень.....	46
Висновки до розділу 6	46
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	51
Додаток А – Інструкція користувача.....	51
Додаток Б. – Технічне завдання на розробку	54
Додаток В. – Екранні форми ІС	63

ВСТУП

Сучасний етап розвитку цифрової економіки характеризується стрімким зростанням обсягів даних та динамічними змінами ринкових процесів. Прогнозування попиту стає одним із ключових елементів ефективного управління підприємствами, оскільки дозволяє оптимізувати виробничі, логістичні та фінансові процеси. Точність прогнозів безпосередньо впливає на рівень прибутковості, мінімізацію витрат та здатність організацій швидко адаптуватися до змін зовнішнього середовища.

Водночас у великих потоках даних можуть виникати аномальні відхилення — різкі зміни поведінки споживачів, сезонні сплески чи падіння попиту, збої в постачанні або форс-мажорні обставини. Виявлення таких аномалій на ранніх етапах дозволяє приймати оперативні рішення, зменшувати ризики та підтримувати стабільність бізнес-процесів. Традиційні статистичні методи не завжди забезпечують необхідну гнучкість та точність у таких умовах.

Сучасні досягнення в галузі машинного навчання (ML) відкривають нові можливості для інтелектуального аналізу даних і побудови адаптивних прогнозних моделей. Алгоритми ML дозволяють не лише моделювати складні закономірності у часових рядах, але й автоматично виявляти аномалії, що значно підвищує ефективність управлінських рішень.

Використання середовища .NET як технологічної платформи забезпечує інтеграцію розроблених інтелектуальних моделей безпосередньо у корпоративні інформаційні системи. Це дозволяє створити масштабоване та продуктивне рішення, придатне для реального бізнес-середовища.

Актуальність обраної теми зумовлена потребою підприємств у точних, адаптивних та автоматизованих інструментах прогнозування попиту, які поєднують аналітичні методи та технології машинного навчання. Такий підхід дозволяє не лише передбачати тенденції розвитку попиту, а й виявляти нетипові ситуації, що впливають на прийняття управлінських рішень.

Об'єкт дослідження є процес прогнозування попиту та виявлення аномальних відхилень у динамічних потоках даних.

Предмет дослідження є методи та інструменти машинного навчання, а також технологічні засоби середовища .NET, що використовуються для побудови моделей прогнозування попиту та виявлення аномалій.

Мета дослідження розробити та реалізувати інтелектуальну систему прогнозування попиту з виявленням аномалій на основі моделей машинного навчання з інтеграцією у середовище .NET для підтримки управлінських рішень.

Завдання дослідження

1. Провести аналіз предметної області прогнозування попиту та виявлення аномалій.
2. Дослідити сучасні методи та алгоритми машинного навчання, що застосовуються для прогнозування часових рядів.
3. Обґрунтувати вибір моделей ML для інтеграції у .NET середовище.
4. Розробити математичну модель процесу прогнозування та виявлення аномалій.
5. Реалізувати програмний модуль у середовищі .NET з використанням ML.NET.
6. Провести експериментальні дослідження ефективності розробленої системи.
7. Оцінити точність прогнозування та якість виявлення аномалій, сформулювати рекомендації щодо практичного застосування.

Гіпотеза дослідження

Якщо застосувати адаптивні методи машинного навчання для аналізу часових рядів попиту та виявлення аномалій, а також інтегрувати ці моделі у середовище .NET, то можна суттєво підвищити точність прогнозування попиту та оперативність виявлення нетипових змін у даних, що забезпечить більш ефективну підтримку управлінських рішень.

Наукова новизна роботи полягає у розробці підходу до поєднання моделей прогнозування попиту та алгоритмів виявлення аномалій у єдиній системі. Інтеграції моделей машинного навчання в середовище .NET із використанням ML.NET для створення інтелектуального модуля прийняття рішень. Удосконаленні процесу прогнозування за рахунок комбінованого використання аналітичних та інтелектуальних методів аналізу часових рядів. Запропонованні способу підвищення точності прогнозів за рахунок адаптивного коригування моделі з урахуванням виявлених аномалій.

Практичне значення одержаних результатів.

Розроблена система може бути використана в організаціях, що займаються плануванням постачання, аутсорсингом, виробництва або логістики.

Інтеграція прогнозних моделей у .NET забезпечує можливість впровадження рішення безпосередньо у корпоративну IT-інфраструктуру.

Отримані результати можуть бути використані для підвищення ефективності управлінських рішень, зниження витрат та мінімізації ризиків.

Розроблені підходи можуть бути адаптовані для інших галузей, де важливим є виявлення аномалій у потоках даних (енергетика, транспорт, роздрібна торгівля тощо)[1].

РОЗДІЛ 1. ПРОГНОЗУВАННЯ ПОПИТУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ ІЗ ВИКОРИСТАННЯМ ML У СЕРЕДОВИЩІ .NET

1.1 Аналіз предметної області

Сучасні підприємства функціонують в умовах високої конкуренції, динамічного ринку та стрімкого розвитку цифрових технологій. Ефективне управління діяльністю організації потребує оперативного прийняття рішень, які базуються на точній, повній та своєчасній інформації. Водночас традиційні підходи до збору та обробки даних часто не забезпечують необхідного рівня глибини аналізу та швидкості реагування на зміни зовнішнього і внутрішнього середовища.

У процесі діяльності підприємства формується великий масив інформації — дані про фінансові показники, запаси, виробництво, продажі, клієнтів, постачальників, персонал тощо. Ці дані містять потенційно цінні знання, але без відповідних методів їх обробки та інтерпретації залишаються невикористаними. Саме технології інтелектуального аналізу даних (Data Mining) дозволяють виявляти приховані закономірності, тренди та взаємозв'язки між показниками діяльності.

Застосування інтелектуальних методів аналізу даних дає змогу виявляти чинники, що найбільше впливають на ефективність управління, прогнозувати ключові показники (виручку, витрати, рентабельність), оптимізувати використання ресурсів, своєчасно виявляти ризики та відхилення від планових значень, підвищити обґрунтованість управлінських рішень [2].

Крім того, важливим аспектом є інтеграція інтелектуального аналізу даних у вже наявні інформаційні системи підприємства. Це дає змогу автоматизувати процес збору, обробки та візуалізації аналітичної інформації, зменшуючи вплив людського фактора та скорочуючи час ухвалення управлінських рішень.

Таким чином, предметна область цієї теми охоплює сукупність методів, засобів та процесів аналізу управлінських, фінансових і операційних даних підприємства з метою підвищення ефективності його функціонування, конкурентоспроможності та стійкості до змін ринкового середовища.

1.2 Аналіз існуючих недоліків

Попит на товари та послуги є одним із ключових чинників ефективності функціонування підприємств у різних галузях економіки. Прогнозування попиту має вирішальне значення для планування виробництва, управління запасами, логістики та стратегічного розвитку. Водночас існуючі підходи до прогнозування мають низку суттєвих недоліків, що знижують точність та ефективність прийняття рішень.

1. Обмежені можливості класичних статистичних методів. Багато підприємств досі використовують традиційні методи прогнозування (ковзне середнє, експоненціальне згладжування, лінійна регресія тощо). Хоча вони є простими у реалізації, такі моделі часто не враховують нелінійність, сезонність та вплив зовнішніх факторів, що призводить до неточних прогнозів.

2. Недостатнє виявлення аномальних даних. Аномалії в даних (пікові навантаження, раптові зростання або падіння попиту) часто ігноруються або сприймаються як випадкові помилки. Це ускладнює побудову коректних моделей та може призводити до спотворення прогнозів. Багато систем не мають вбудованих механізмів автоматичного виявлення та обробки таких відхилень.

3. Відсутність адаптивності моделей.

У мінливих ринкових умовах моделі, що не оновлюються автоматично, швидко втрачають актуальність. У більшості традиційних систем прогнозування не передбачено механізмів самооновлення або адаптації до нових даних у режимі реального часу.

4. Фрагментарність програмних рішень.

Існуючі інструменти часто реалізовані у вигляді окремих аналітичних модулів, що ускладнює їх інтеграцію з іншими бізнес-процесами та корпоративними системами. Відсутність єдиного середовища для прогнозування, аналітики та управління знижує ефективність використання таких рішень.

5. Недостатня інтеграція з корпоративними платформами.

Попри розвиток сучасних технологій, у багатьох компаніях прогностичні модулі не інтегровані у робоче середовище розробки, таке як .NET. Це ускладнює їх масштабування, автоматизацію процесів та підтримку стабільності роботи на промисловому рівні.

6. Відсутність єдиної методології для побудови моделей.

У практиці підприємств нерідко використовуються різні алгоритми прогнозування без урахування специфіки галузі, структури даних чи бізнес-процесів. Це призводить до непослідовності результатів і знижує довіру до систем прогнозування [6].

7. Недостатня підтримка користувачів без технічних знань.

Існуючі рішення часто орієнтовані на аналітиків або програмістів, що ускладнює їх використання менеджерами середньої ланки. Відсутність зручного інтерфейсу, автоматичних налаштувань та візуалізації знижує ефективність практичного застосування таких систем [3].

1.3 Варіанти вирішення існуючих недоліків та їх порівняльний аналіз

Виявлені у попередньому підпункті недоліки традиційних систем прогнозування попиту та виявлення аномалій вказують на необхідність використання сучасних підходів, які забезпечують адаптивність, високу точність, інтегрованість та зручність експлуатації. У межах цього дослідження розглянуто два альтернативних варіанти вирішення вказаних проблем.

Варіант 1. Інтеграція готових бібліотек машинного навчання у середовище .NET

Цей підхід передбачає використання існуючих рішень, таких як ML.NET — нативна бібліотека машинного навчання для .NET, TensorFlow.NET — адаптація популярної ML-платформи для .NET, ONNX Runtime — для розгортання попередньо натренованих моделей.

Переваги

1. Простота інтеграції з існуючими корпоративними системами на .NET;
2. Висока продуктивність завдяки нативній підтримці платформи;
3. Можливість швидкої розробки MVP або промислових рішень;
4. Підтримка автоматизованого навчання моделей, крос-валідації, детекції аномалій та прогнозування часових рядів [4].

Недоліки

1. Обмежена гнучкість порівняно з кастомними моделями;
2. Складність реалізації специфічних алгоритмів, що не входять до стандартного набору бібліотек;
3. Потреба у високій кваліфікації розробника для налаштування та оптимізації моделей.

Варіант 2. Розробка гібридного рішення з використанням зовнішніх ML-сервісів через API

У цьому варіанті для прогнозування та виявлення аномалій використовуються зовнішні сервіси (наприклад, Azure Machine Learning, Google Cloud Vertex AI чи Amazon SageMaker), а результати інтегруються у .NET-додаток через REST API.

Переваги.

1. Широкий набір готових алгоритмів та інструментів;
2. Автоматичне масштабування, моніторинг та керування моделями;
3. Можливість використання складних моделей без необхідності їх локального навчання;
4. Висока точність завдяки використанню передових ML-технологій.

Недоліки

1. Залежність від зовнішніх постачальників сервісів та інтернет-з'єднання;
2. Можливі додаткові витрати на хмарну інфраструктуру;
3. Складність забезпечення безпеки та конфіденційності даних;
4. Залежність від форматів і оновлень зовнішніх API [5].

Порівняльна характеристика варіантів

Критерій	Варіант 1: ML.NET (вбудоване)	Варіант 2: Зовнішні ML- сервіси через API
Інтеграція	Висока (рідна підтримка .NET)	Середня (через REST API)
Гнучкість моделей	Середня	Висока
Вартість	Низька (локальне рішення)	Висока (залежно від сервісу)
Продуктивність	Висока	Висока (залежить від інфраструктури провайдера)
Залежність від інфраструктури	Низька	Висока
Безпека даних	Висока (локальні обчислення)	Середня (залежить від політики постачальника)
Можливість масштабування	Середня	Висока
Час впровадження	Короткий	Середній або довгий

Обидва варіанти мають свої переваги та обмеження.

Варіант 1 доцільно використовувати для підприємств, які прагнуть мати автономну, безпечну та швидку систему прогнозування попиту з виявленням аномалій.

Варіант 2 є доцільним, коли потрібна висока точність прогнозів, гнучкість і масштабованість, навіть за рахунок залежності від зовнішніх сервісів.

У межах магістерського дослідження може бути реалізовано один із цих варіантів або розроблено гібридний підхід, який поєднує локальну обробку даних з підключенням до зовнішніх ML-сервісів для підвищення точності прогнозів.

1.4. Концепція системи

Обраний варіант 2 передбачає побудову гібридної системи прогнозування попиту та виявлення аномалій із використанням моделей машинного навчання, які розгортаються у вигляді окремих сервісів і інтегруються з бізнес-додатком через API у середовищі .NET. Така концепція доцільна у випадках, коли пріоритетом є висока точність прогнозів, масштабованість та гнучка архітектура, навіть за умови залежності від зовнішніх або спеціалізованих ML-сервісів.

Основна ідея концепції

Система розділяється на два основні рівні:

ML-рівень (аналітичне ядро) — відповідає за прогнозування попиту та виявлення аномалій, працює на основі сучасних моделей машинного навчання (наприклад, Long short-term memory, Gradient boosting чи Prophet).

Бізнес-додаток у .NET — відповідає за обробку вхідних даних, візуалізацію результатів, інтеграцію з CRM, ERP або іншими корпоративними системами.

ML-моделі можуть бути розгорнуті:

як окремий мікросервіс у контейнерах Docker;

у вигляді веб API (наприклад, через FastAPI чи Flask);

у хмарних середовищах, зокрема на платформах Azure Machine Learning чи Amazon SageMaker.

Принцип роботи системи

Користувач у бізнес-додатку ініціює прогнозування або аналіз історичних даних.

Дані з БД передаються до ML-сервісу через REST або gRPC API.

Модель обробляє вхідні дані та формує:

прогноз попиту на визначений період;

список потенційних аномалій.

Результати повертаються у .NET-додаток для відображення в аналітичній панелі або передачі в інші модулі системи.

При необхідності модель автоматично оновлюється з урахуванням нових даних (адаптивне навчання).

Переваги концепції

Висока точність — завдяки використанню складних моделей ML та можливості їх регулярного перенавчання.

Масштабованість — легко адаптується до збільшення обсягів даних та кількості користувачів.

Гнучкість — можна використовувати різні моделі, не змінюючи базову архітектуру системи.

Простота інтеграції — через API можна підключати зовнішні сервіси або власні ML-моделі без глибокого втручання у код бізнес-додатку.

Недоліки концепції

Залежність від зовнішніх сервісів — у разі збою або зміни політики доступу можуть виникнути затримки чи проблеми з доступністю.

Більша складність адміністрування — потребує налаштування контейнерів, мережевої безпеки, моніторингу сервісів.

Підвищені вимоги до безпеки — оскільки дані можуть передаватися між різними середовищами [6].

Таким чином, концепція на основі варіанта 2 є оптимальною для компаній, які працюють з великими обсягами даних, мають динамічний ринок і потребують точних прогнозів у режимі реального часу. Її гнучкість і масштабованість роблять систему придатною для впровадження в

аутсорсингових компаніях, які розробляють комплексні Full Stack .NET рішення для бізнесу.

1.5 Математичне забезпечення розробки

Математичне забезпечення розробки системи прогнозування попиту та виявлення аномалій

Математичне забезпечення системи прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET є ключовим елементом, що визначає точність і ефективність функціонування всієї аналітичної платформи. Воно включає формалізацію задачі, вибір та адаптацію математичних моделей, алгоритмів машинного навчання, а також методів обробки та аналізу даних[7].

1. Формалізація задачі прогнозування

Нехай

$$D = \{(x_t, y_t) \mid t = 1, 2, \dots, T\}$$

— це історичний набір даних, де x_t — вектор незалежних змінних (факторів), які впливають на попит (сезонність, тренди, ціна, маркетингові активності, економічні показники тощо), а y_t — значення попиту в момент часу t .

Задача прогнозування полягає у знаходженні функції:

$$\hat{y}_{t+1} = f(x_t, x_{t-1}, \dots, x_{t-k})$$

де $\hat{y}_{t+1}$ — прогнозоване значення попиту, а k — глибина врахованої історії.

Мета — мінімізувати похибку прогнозування

$$\min \mathcal{L} = \frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2$$

де $\mathcal{L}$ — функція втрат типу MSE (Mean Squared Error).

Математична модель виявлення аномалій

Для виявлення аномалій використовується підхід порівняння реального значення попиту y_t з прогнозованим $\hat{y}_t$. Аномалія фіксується, якщо

$$|y_t - \hat{y}_t| > \tau$$

де τ — порогове значення, що визначається статистично (наприклад, як $\mu + 3\sigma$ для нормального розподілу).

Також можуть бути застосовані алгоритми машинного навчання, орієнтовані на виявлення аномалій, такі як:

кластеризація з подальшою перевіркою відстаней (K-Means, DBSCAN),
метод локальної щільності (LOF),
ізоляційні дерева (Isolation Forest).

3. Алгоритмічна база

Серед моделей машинного навчання, які можуть бути реалізовані:

Регресійні моделі — для базового прогнозування попиту;

ARIMA та Prophet — для часових рядів;

Long short-term memory (LSTM) — для врахування довгострокових залежностей;

Gradient boosting — для побудови високоточних моделей на структурованих даних;

Random forest — для комбінованих факторів впливу.

Для моделі LSTM прогнозна функція має вигляд:

$$\hat{y}_{t+1} = F_{\text{LSTM}}(y_t, y_{t-1}, \dots, y_{t-n})$$

де F_{LSTM} — це функція рекурентної нейронної мережі.

4. Оптимізація параметрів

Для мінімізації похибки прогнозу та підвищення точності виявлення аномалій використовуються методи оптимізації:

стохастичний градієнтний спуск (SGD);
адаптивні методи (Adam, RMSprop);
крос-валідація для уникнення перенавчання.

$$\theta^* = \arg \min_{\theta} \mathcal{L}(f_{\theta}(x), y)$$

де θ — параметри моделі.

5. Математичне забезпечення інтеграції в .NET

У середовищі .NET математичні моделі реалізуються через:
вбудовані бібліотеки ML.NET;
імпорт попередньо натренованих моделей (ONNX, TensorFlow);
побудову сервісів прогнозування та аномалій на основі API.

Система підтримує:

$$\text{Predict}(X) \rightarrow \hat{Y}, \quad \text{AnomalyDetect}(\hat{Y}, Y) \rightarrow \text{Flag}$$

де Flag — індикатор аномального значення.

Таким чином, математичне забезпечення системи охоплює комплекс методів часових рядів, машинного навчання та оптимізаційних алгоритмів, що у поєднанні з можливостями .NET створює гнучку та точну платформу для прогнозування попиту та автоматичного виявлення аномальних відхилень у динаміці продажів [8].

1.6 Постановка задачі

В умовах високої конкуренції та динамічних змін на ринку IT-послуг аутсорсингові компанії, що спеціалізуються на розробці Full Stack .NET рішень, потребують ефективних інструментів для оптимізації процесів планування ресурсів та управління проєктами. Одним із ключових аспектів такого управління є точне прогнозування попиту на послуги та виявлення аномальних відхилень у динаміці навантаження та споживання ресурсів.

Сучасні тенденції показують, що класичні статистичні методи часто не забезпечують належного рівня точності та адаптивності. Це обумовлює

необхідність впровадження інтелектуальних методів, зокрема технологій машинного навчання (ML), у середовище .NET, яке вже є базовою технологічною платформою компанії. Таке рішення дозволить не лише підвищити якість прогнозування, а й інтегрувати аналітичні інструменти безпосередньо у вже існуючі бізнес-додатки.

Основна ціль задачі розробити інтелектуальну систему прогнозування попиту та виявлення аномалій, інтегровану у .NET-екосистему аутсорсингової компанії, яка забезпечує високу точність прогнозів на основі часових рядів та факторного аналізу, автоматично виявляє та сигналізує про відхилення від нормального попиту (аномалії), інтегрується з існуючими Full Stack .NET застосунками через API або внутрішні модулі, підтримує масштабування та адаптацію до зростання обсягів даних.

Вихідні умови

Наявність історичних даних про навантаження на ресурси (кількість проєктів, кількість активних розробників, години роботи, запити клієнтів, бюджет тощо).

Доступ до корпоративних систем, розроблених на .NET.

Можливість розгортання ML-моделей у вигляді окремих мікросервісів або локально інтегрованих компонентів. Стандартизований API для взаємодії з іншими модулями компанії [9].

Необхідно розв'язати такі завдання проаналізувати та підготувати історичні дані для моделювання попиту. Обрати та налаштувати ефективні ML-моделі для прогнозування часових рядів. Реалізувати механізм автоматичного виявлення аномалій на основі статистичних та машинних методів. Забезпечити інтеграцію моделі у середовище .NET з можливістю масштабування. Розробити інтерфейс або API для відображення прогнозів та попереджень. Оцінити ефективність моделі за ключовими метриками точності та продуктивності.

Очікуваний результат

У результаті виконання поставленої задачі буде створено інтелектуальний аналітичний модуль, який дозволить компанії ефективніше планувати людські та технічні ресурси, швидше реагувати на зміни попиту, зменшити ризики перевантаження або простоїв, покращити управлінські рішення завдяки аналітиці в реальному часі.

Таким чином, постановка задачі орієнтована не лише на технічну реалізацію прогнозної моделі, а й на її інтеграцію в бізнес-процеси компанії, що забезпечує реальну комерційну цінність рішення [10].

Висновки до розділу 1

1. У першому розділі було розглянуто загальну концепцію та обґрунтування необхідності розробки системи прогнозування попиту та виявлення аномалій із використанням технологій машинного навчання у середовищі .NET для аутсорсингової компанії, що спеціалізується на Full Stack .NET розробці.

2. Визначено, що класичні методи аналітики не забезпечують достатньої гнучкості та точності для оперативного реагування на коливання ринкового попиту та внутрішні зміни навантаження. Зокрема, вони не дозволяють своєчасно виявляти аномалії, які можуть сигналізувати про ризики — наприклад, різке збільшення або спад запитів клієнтів, незбалансоване використання ресурсів чи нестандартну поведінку користувачів.

3. Було проведено аналіз предметної області, що охоплює специфіку управління попитом та ресурсами в ІТ-аутсорсинговій компанії, особливості інтеграції прогнозних моделей у корпоративну архітектуру, а також визначено ключові вхідні дані та фактори, що впливають на попит. Встановлено, що застосування моделей машинного навчання — таких як Long short-term memory, Gradient boosting чи Prophet — дозволяє істотно підвищити якість прогнозування.

4. Сформульовано постановку задачі, яка передбачає розробку інтелектуального модуля, здатного прогнозувати попит на основі часових

рядів і факторного аналізу, виявляти та сигналізувати про аномалії, інтегруватися з існуючими .NET-рішеннями компанії через API чи мікросервісну архітектуру.

5. Також визначено основні вимоги до системи, вихідні умови, завдання та очікувані результати. Обґрунтовано доцільність використання гібридної архітектури з ML-компонентом як окремим сервісом, що забезпечує високу точність, масштабованість і гнучкість.

Таким чином, у першому розділі створено міцне методологічне та концептуальне підґрунтя для подальшої розробки математичного забезпечення та архітектури системи, яка буде представлена в наступних розділах роботи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІС ПРОГНОЗУВАННЯ ПОПИТУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ ІЗ ВИКОРИСТАННЯМ ML У СЕРЕДОВИЩІ .NET В АУТСОРСИНГОВІЙ КОМПАНІЇ, ЩО СПЕЦІАЛІЗУЄТЬСЯ НА РОЗРОБЦІ FULL STACK .NET РІШЕНЬ

2.1 Загальна архітектура системи прогнозування

Проектowana система призначена для автоматизації процесу прогнозування попиту на IT-послуги та виявлення аномалій у завантаженні ресурсів аутсорсингової компанії, яка спеціалізується на full stack.NET рішеннях. Основна ідея полягає у створенні модульного прогнозного пайплайну, який інтегрується безпосередньо у внутрішню корпоративну інформаційну систему компанії.

Система складається з таких основних компонентів, як модуль збору та підготовки даних — відповідає за отримання історичних даних про завантаження проєктних команд, попит на послуги, кількість активних контрактів та проєктів.

Модуль обробки та очищення даних — виконує нормалізацію, агрегацію та трансформацію даних у формат, придатний для машинного навчання.

Модуль навчання ML-моделі — реалізований з використанням бібліотеки ML.NET, що дозволяє будувати моделі часових рядів і виявляти аномалії.

Модуль прогнозування та детекції аномалій — генерує прогнози майбутнього попиту та сигналізує про відхилення від тренду.

Модуль інтеграції з .NET-додатком — забезпечує передачу результатів у вигляді API або безпосередньо в UI системи управління ресурсами.

Модуль візуалізації — інтерактивні графіки та дашборди для менеджерів [11].

2.2 Загально системні рішення

User / CRM ініціює запит через інтерфейс. API Gateway передає запит у PredictionService. Сервіс отримує історичні дані з БД. Виконується виклик

ML-моделі через ML.NET або ONNX Runtime. Результат зберігається у базі даних і повертається у вигляді готового прогнозу. Інтерфейс візуалізує результати та сигнали про аномалії.

Пояснення компонентів Presentation Layer — інтерфейс користувача або дашборд, де відображаються прогнози. Application Layer — бізнес-логіка на базі .NET маршрутизація, виклики сервісів, планування задач. ML Layer — окремий модуль із навченою моделлю, який виконує прогнозування та виявлення аномалій. Data Layer — зберігання історичних даних та ETL-процеси їх попередньої обробки див.рис. 2.1.

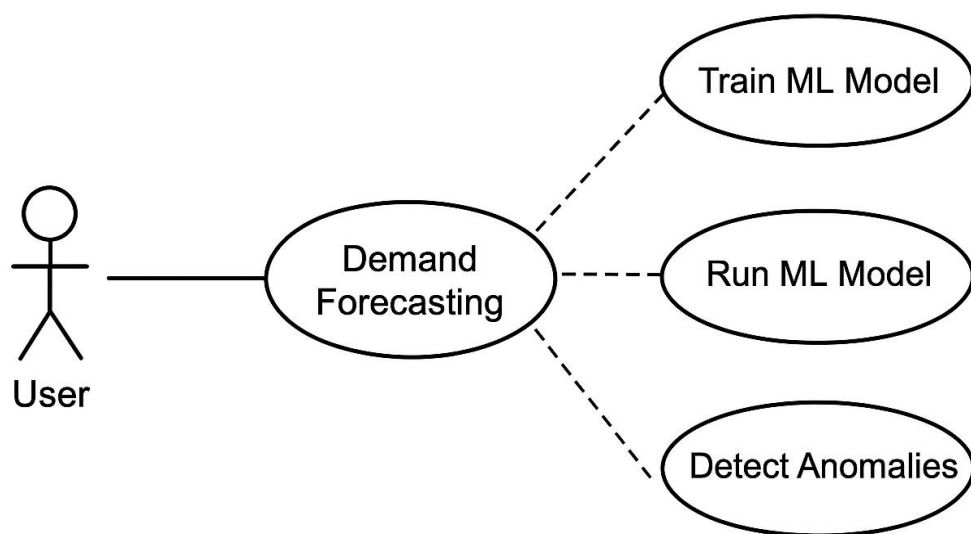


Рисунок 2.1 – Діаграма варіантів використання

2.3 Налаштування пайплайну прогнозування

Для побудови прогнозного пайплайну використовується вбудований інструментарій ML.NET. Процес налаштування складається з кількох етапів.

Крок 1. Збір і підготовка даних

Підключення до корпоративної бази даних через ORM (наприклад, Entity Framework Core). Завантаження історичних даних про кількість активних проєктів, залучені ресурси та фінансові показники.

Формування датасету у вигляді часових рядів.

Крок 2. Попередня обробка

Виявлення та видалення пропусків у даних. Нормалізація значень для зменшення впливу масштабів. Розбиття даних на тренувальний та тестовий набори [12].

Крок 3. Створення моделі

Використання Time Series Forecasting API бібліотеки ML.NET для прогнозування попиту. Налаштування параметрів моделі (вікно прогнозу, період, тренди). Додавання модуля Anomaly Detection для виявлення відхилень від тренду.

Крок 4. Навчання та оцінювання

Навчання моделі на тренувальних даних. Обчислення точності прогнозів за метриками RMSE, MAPE та R^2 . Збереження моделі у форматі .zip для подальшого використання.

Крок 5. Прогнозування та виявлення аномалій

Завантаження натренованої моделі у продакшн-додаток. Автоматичне генерування прогнозів на обраний період (наприклад, 2–4 тижні вперед). Детекція пікових відхилень (аномалій), які сигналізують про перевантаження або недовантаження ресурсів [13].

2.3 Інтеграція в .NET-додаток

Інтеграція виконується безпосередньо у back-end частину корпоративного додатку. Створення окремого сервісу ForecastingService використовується Dependency Injection для гнучкої архітектури. Сервіс завантажує модель ML.NET і виконує прогноз при кожному оновленні даних.

Реалізація API включає REST-контролер (на основі ASP.NET Core) надає кінцеві точки для отримання прогнозів у вигляді JSON. Інші модулі системи можуть використовувати ці дані для планування.

Інтеграція з UI це коли дані прогнозів виводяться у вигляді інтерактивних графіків (через Blazor або JavaScript-бібліотеки, наприклад Chart.js).

Аномальні значення підсвічуються в інтерфейсі червоним кольором для швидкої реакції менеджера.

2.4 Механізм автоматизації та оновлення моделі

Модель може автоматично перевчатися раз на тиждень або місяць на основі нових даних. Для цього використовується планувальник завдань Windows або фонові сервіси .NET (IHostedService). Передбачено логування результатів прогнозування та аномалій у окрему таблицю для подальшого аналізу [14].

2.5 Очікувані результати впровадження

Зниження ризиків перевантаження команд та неефективного планування. Підвищення точності прогнозування попиту на послуги аутсорсингової компанії.

Прискорення процесу ухвалення управлінських рішень за рахунок автоматизованих дашбордів. Збільшення прибутковості проєктів завдяки оптимізації розподілу ресурсів.

Запропонований підхід дозволяє поєднати машинне навчання з бізнес-процесами компанії, не виходячи за межі звичного технологічного стеку .NET. Така архітектура легко масштабована, безпечна та не потребує значних витрат на інтеграцію сторонніх сервісів. Налаштування пайплайну прогнозування

Для автоматизації процесу прогнозування попиту та виявлення аномалій у вибраному аутсорсинговому середовищі розроблено гнучкий ML-пайплайн, який інтегрується безпосередньо в .NET-додаток. Архітектура побудована за принципом модульності та масштабованості, що дозволяє легко оновлювати або замінювати окремі компоненти без впливу на загальну систему.

Пайплайн передбачає такі етапи як збір та попередня обробка даних, підключення до внутрішніх джерел даних (CRM, ERP, бази продажів), очистка даних від пропусків, дублювань, аномальних значень, агрегація за часовими інтервалами [15].

Формування ознак (Feature Engineering) в створенні часових лагів та ковзних середніх, виявлення сезонних закономірностей, додавання бізнес-факторів (акції, маркетингові кампанії тощо).

Навчання та тестування моделі ML це використання моделей часового ряду (наприклад Prophet або LSTM), побудова моделі аномалій (наприклад, Isolation Forest або Autoencoder), крос-валідація та оптимізація гіперпараметрів.

Деплой та інтеграція в .NET-додаток експортування моделі у формат ONNX, підключення до бізнес-логіки через API або сервіс передбачень, реалізація виклику моделі у відповідь на події користувачів чи cron-задачі.

Відображення результатів і прийняття рішень в формуванні прогнозу на визначений період, сигналізація при виявленні аномалій, інтеграція з панеллю моніторингу та дашбордами.

інтеграція з .NET-додатком

Для впровадження пайплайну в інфраструктуру .NET використано шар сервісів прогнозування, який надає REST API для взаємодії з фронтендом та іншими модулями.

Основні кроки інтеграції підключення бібліотек машинного навчання через ML.NET або через ONNX Runtime. Створення сервісу PredictionService для обробки запитів. Налаштування DI-контейнера для реєстрації сервісів. Використання BackgroundService для планового оновлення моделей. Вивід прогнозів на UI (Blazor / React + .NET API) [16].

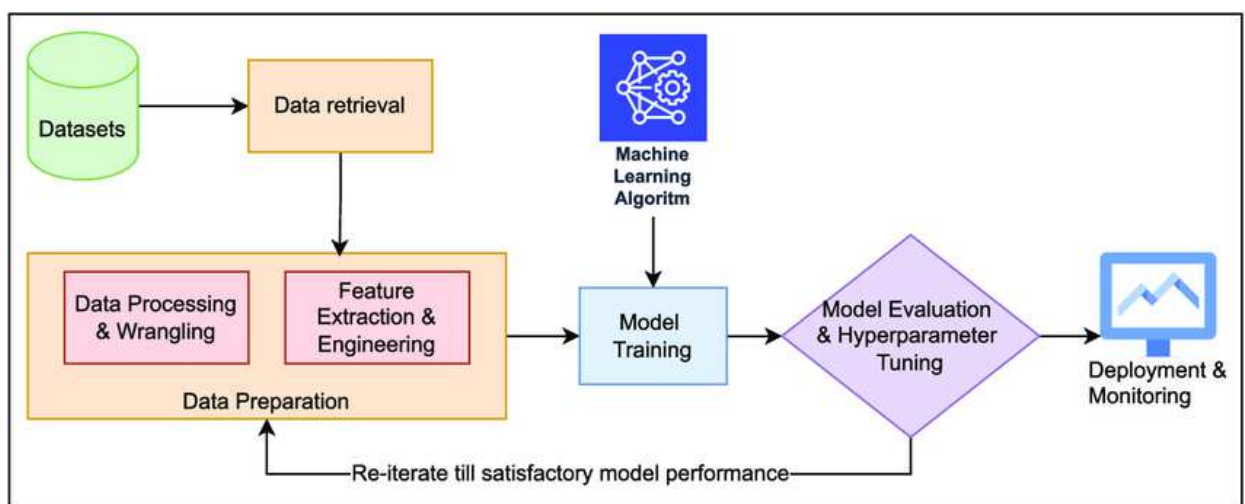


Рисунок 2.2 - Опис послідовності викликів

Користувач / CRM-система → надсилає запит до API прогнозування.
API Gateway (.NET) → викликає Prediction Service. Prediction Service →

обробляє дані, викликає ML Engine (ONNX / ML.NET). ML Engine → повертає прогноз та інформацію про аномалії. Prediction Service → формує відповідь та зберігає результат у БД. UI / Дашборд → візуалізує результати для бізнес-користувачів.

Переваги такої архітектури полягають в повній інтеграція з наявними бізнес-додатками .NET, рис.2.2, можливість оновлення моделі без зупинки системи, масштабованість та підтримка CI/CD, зручна діагностика та моніторинг прогнозів.

Можливі покращення це додавання черг обробки даних (наприклад, Apache Kafka), підтримка кількох моделей для різних сегментів попиту, вбудований модуль адаптивного навчання в реальному часі, див рис.2.3.

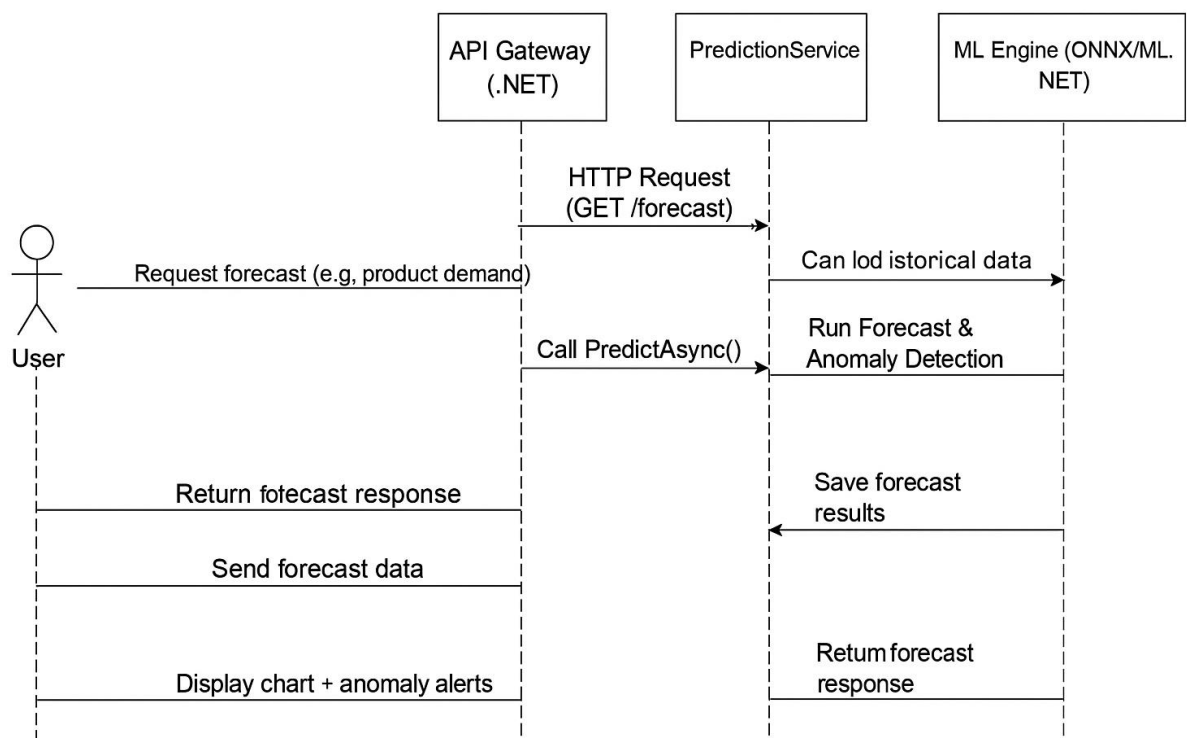


Рисунок 2.3 – Діаграма послідовності

2. 6 Проектування бази даних розробки

База даних у системі прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET є ключовим компонентом, оскільки

забезпечує зберігання історичних даних про продажі, запаси, клієнтів, ціни та інші параметри, необхідні для побудови моделей прогнозування.

Організацію доступу до даних для алгоритмів машинного навчання, які потребують великих обсягів якісної інформації для навчання.

Цілісність і структурованість інформації, що дозволяє забезпечити узгодженість результатів прогнозів та аналітичних висновків.

Можливість інтеграції з .NET-застосунком, ML-сервісами та інтерфейсом користувача через API.

Ефективне оновлення та обробку даних у реальному часі, що є критично важливим для виявлення аномалій і швидкого реагування на зміни попиту [17].

Таким чином, база даних виступає фундаментом системи, на якому базується як аналітична, так і прогнозна функціональність.

Логічна модель визначає структуру даних і взаємозв'язки між сутностями, без урахування фізичної реалізації. Основні сутності системи.

Products (Товари)

ProductID (PK)

Name

Category

Price

SupplierID (FK)

Suppliers (Постачальники)

SupplierID (PK)

Name

ContactInfo

Sales (Продажі)

SaleID (PK)

ProductID (FK)

Date

Quantity

TotalAmount

Forecasts (Прогнози)

ForecastID (PK)

ProductID (FK)

ForecastDate

PredictedDemand

ConfidenceLevel

Anomalies (Аномалії)

AnomalyID (PK)

ProductID (FK)

DetectionDate

AnomalyType

Severity

Зв'язки між таблицями.

Products —< Sales

Products —< Forecasts

Products —< Anomalies

Suppliers —< Products

Фізична модель бази даних

Фізична модель, див.рисунок 2.4, визначає реалізацію таблиць у СУБД (Microsoft SQL Server). У ній уточнюються типи даних, індекси, зовнішні ключі тощо.

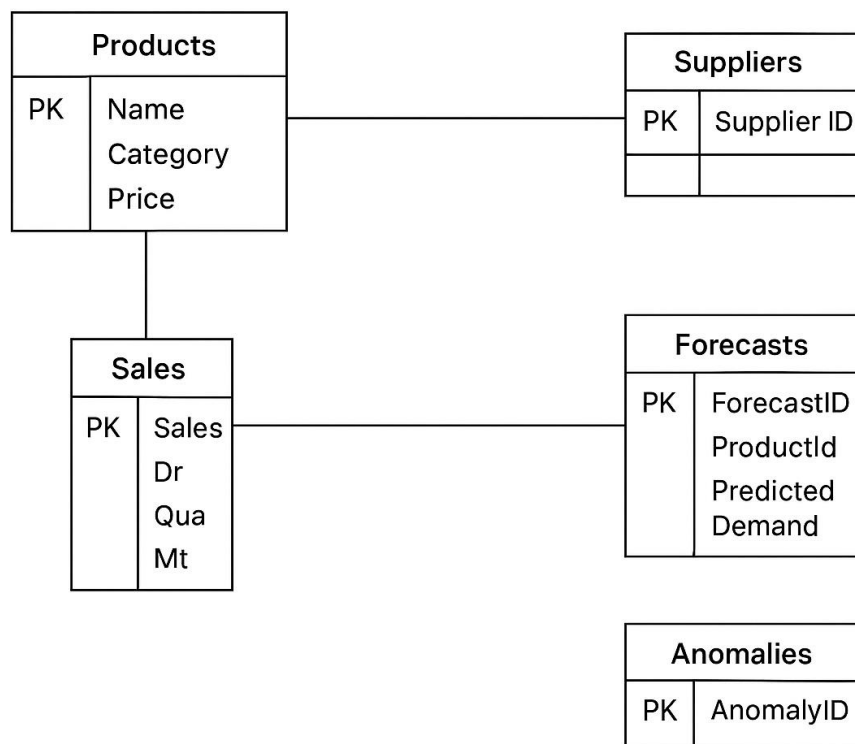


Рисунок 2.4 – Фізична модель БД

Висновки до розділу 2

У другому розділі було здійснено комплексне проєктування інформаційної системи, призначеної для автоматизованого прогнозування попиту та виявлення аномалій на основі методів машинного навчання в середовищі .NET. У процесі проєктування сформовано архітектурну модель системи, визначено основні функціональні модулі, інтерфейси взаємодії та програмно-апаратні вимоги до розроблюваного рішення.

На основі аналізу предметної області обґрунтовано вибір мікросервісної архітектури, що забезпечує масштабованість, гнучкість і можливість незалежного оновлення окремих компонентів. Визначено ключові підсистеми — модуль збору та первинної обробки даних, модуль аналітики та ML-обробки, модуль виявлення аномалій, підсистема прогнозування, система зберігання даних та інтерфейс користувача. Для побудови моделей прийнято рішення про використання можливостей ML.NET, що дозволяє інтегрувати

алгоритми машинного навчання безпосередньо у .NET-середовище та спростити деплой.

Розроблено логічну та фізичну структуру бази даних, визначено інформаційні потоки, формати даних та механізми їх передачі між компонентами системи. Особлива увага приділена забезпеченню якості даних, оскільки коректність моделей прогнозування безпосередньо залежить від повноти, цілісності та достовірності вхідної інформації.

Проведено вибір оптимальних алгоритмів для вирішення завдань прогнозування (регресійні моделі, дерева рішень, Time Series Forecasting) та виявлення аномалій (Isolation Forest, One-Class SVM), а також розглянуто механізми їх навчання та переоцінки в реальному середовищі експлуатації. Обґрунтовано використання API-підходу для зв'язку між модулями та застосування контейнеризації для підвищення переносимості системи.

У результаті проєктування отримано цілісну модель інформаційної системи, яка забезпечує інтегроване вирішення задач прогнозування попиту та автоматичної детекції аномалій. Сформовані технічні, функціональні та архітектурні рішення створюють основу для подальшої реалізації та впровадження програмного забезпечення, що відповідає сучасним вимогам продуктивності, надійності та масштабованості.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ ПОПИТУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ ІЗ ВИКОРИСТАННЯМ ML У СЕРЕДОВИЩІ .NET

3.1 Діаграма розгортання системи

Аби показати фізичне/платформне розгортання ключових компонентів системи клієнтські інтерфейси, API/веб-застосунок (.NET), сервіс хостингу ML-моделей (ML.NET / ONNX), сховища даних, брокер повідомлень, моніторинг та платформа розгортання (Kubernetes / PaaS).

Основні вузли (nodes) і компоненти це Clients / Users, Web UI (SPA) — браузер, Admin UI / Mobile app, Edge / Network, API Gateway / Load Balancer (HTTPS, TLS termination, rate limiting, auth), Application Layer, .NET Web Application (ASP.NET Core), REST API, бізнес-логіка, авторизація, виклики до ML сервісу, ML Service (Model Host), ML.NET або контейнер з ONNX Runtime, Мікросервіс для inferencing (передбачення) та детекції аномалій, Може бути окремим microservice або інтегрованим в .NET Web App, Integration / Ingestion, Scheduler / ETL jobs — збір зовнішніх даних (API, CSV, стріми), Message Broker (RabbitMQ / Azure Service Bus / Kafka) — події, черги, стріми даних, Storage, Operational DB: SQL Server / Azure SQL (транзакційні дані, налаштування), Time-series DB: InfluxDB / Timescale (метрики, часові ряди для прогнозування), Object Storage: Azure Blob / AWS S3 (збереження моделей, артефактів, сирих даних), Monitoring & Logging, Prometheus (метрики) + Grafana (дашборди), ELK / EFK або Azure Monitor (логи, трасування), Deployment Platform, Kubernetes cluster (AKS / EKS / GKE) або Azure App Service / Azure Container Instances, Persistent Volumes для БД/Blob (якщо on-prem) [18].

CI/CD - GitHub Actions / Azure DevOps (build → тест → deploy), External Systems, Third-party Data Providers (маркет-дані, ERP), Auth provider (OAuth / IdentityServer / Azure AD).

Стандартні зв'язки клієнт → API Gateway це HTTPS/REST (443), API Gateway → .NET App це HTTP/gRPC .NET App → ML Service: gRPC/HTTP

(для inference).NET App / ML Service → DBs: TCP на порт БД (SQL: 1433 тощо), Ingestion → Message Broker: AMQP / Kafka protocol, Monitoring ← всі сервіси: scrape / push (Prometheus, metrics endpoints), CI/CD → Deployment Platform: kubectl/helm / app service deploy.

Нефункціональні аспекти такі як масштабованість Web App і ML Service — горизонтально масштабовані репліки.

Надійність включає резервне копіювання БД, replicated Time-series DB.

Безпека за рахунок TLS, RBAC, секрети в Vault / Kubernetes Secrets.

Обсяг ресурсів для PoC — 2 CPU / 4GB RAM на сервіс, для прод — виділення GPU (якщо потребує inference з великими моделями) або кілька CPU реплік.

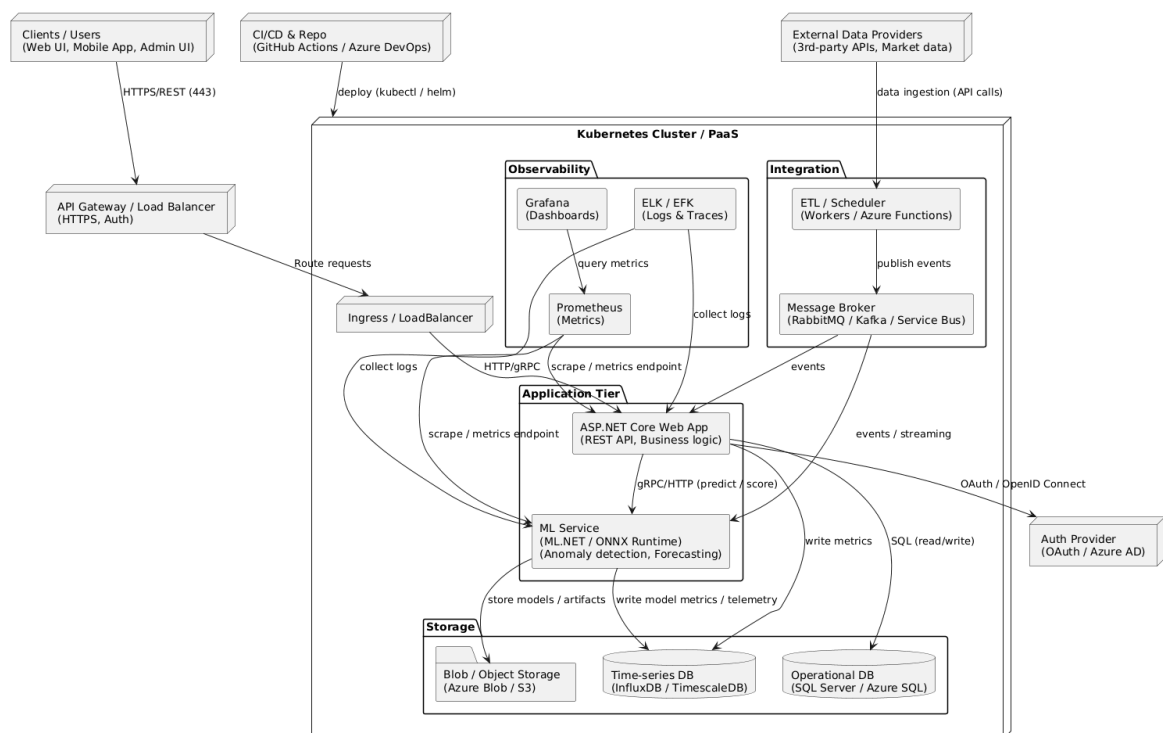


Рисунок 3.1 – Діаграма розгортання системи

3.2 Програмне забезпечення розробки

Програмне забезпечення (ПЗ) призначене для автоматизації процесів збору, обробки, аналізу історичних даних про продажі та зовнішні фактори, з метою побудови прогнозів попиту і виявлення аномалій у поведінці даних.

Розробка реалізується у середовищі .NET 8 із використанням ML.NET для реалізації алгоритмів машинного навчання та ASP.NET Core для створення вебінтерфейсу і REST API.

Система орієнтована на аналітиків, менеджерів із продажу, спеціалістів із планування запасів та розробників, які мають потребу в інструменті для прийняття управлінських рішень на основі даних.

Імпорт та обробка даних зчитування історичних даних продажів у форматах CSV, Excel, JSON, SQL. Автоматичне очищення, нормалізація та агрегація даних. Підтримка завантаження зовнішніх факторів (сезонність, ціни, рекламні кампанії).

Аналітичний модуль включає побудову прогнозів попиту за допомогою алгоритмів ARIMA, Prophet, LSTM (через ONNX/ML.NET integration).

Виявлення аномалій у часових рядах методами Isolation Forest, STL Decomposition, Statistical Z-score. Візуалізація прогнозів та аномалій на графіках (через Chart.js / Plotly).

ML-процеси це навчання моделей на історичних даних. Збереження моделей у форматі .zip / .onnx. Оцінювання точності моделей (MAE, RMSE, MAPE). Можливість оновлення моделей при надходженні нових даних.

Інтерфейс користувача вебінтерфейс (React/Blazor) для завантаження даних, вибору моделі та перегляду результатів. Адміністративна панель для керування користувачами та налаштувань. Експорт результатів прогнозу у форматах Excel / PDF / JSON.

API-сервіси REST API для доступу до функцій прогнозування. Ендпойнти для інтеграції з ERP/CRM системами.

Звітування та журналювання формування звітів про динаміку попиту, точність прогнозів. Збереження логів подій (успішні запити, помилки, тренування моделей).

Прогноз для одного товару за 12 місяців має генеруватись не довше 5 секунд. Підтримка одночасної роботи не менше 10 користувачів. Обробка наборів даних обсягом до 500 000 записів без втрати стабільності.

Масштабованість системи повинна підтримувати розгортання у хмарному середовищі (Azure, AWS, GCP). Компоненти повинні бути контейнеризовані (Docker) з можливістю оркестрації (Kubernetes). Підтримка горизонтального масштабування сервісів.

Забезпечення резервного копіювання бази даних. Обробка помилок з повідомленнями користувачу. Збереження історії прогнозів та версій моделей.

Авторизація та автентифікація користувачів через JWT / OAuth2. Захищений обмін даними через HTTPS. Контроль доступу на основі ролей (Admin, Analyst, Viewer).

Підтримка роботи у браузерях Chrome, Edge, Firefox, Safari. Інтеграція з зовнішніми джерелами даних через REST API або SQL-конектори. Можливість розгортання на Windows Server або Linux.

Інтуїтивний інтерфейс із візуалізацією прогнозів. Довідкова система та підказки. Підтримка української та англійської мов інтерфейсу.

Використання контейнерів Docker для спрощеного розгортання. Сумісність із середовищами Visual Studio, Rider, VS Code.

Код структурується з дотриманням принципів SOLID, Clean Architecture. Оновлення моделей ML виконуються через API без зупинки системи. Модульна архітектура, див табл. 3.1, дозволяє додавати нові алгоритми або джерела даних.

Компонент	Вимога
ОС сервера	Windows Server 2022 / Linux Ubuntu 22.04
Розробницьке середовище	Microsoft Visual Studio 2022 / .NET SDK 8.0
База даних	Microsoft SQL Server / PostgreSQL
Середовище ML	ML.NET, ONNX Runtime
Вебсервер	Kestrel / IIS / Nginx
Контейнеризація	Docker, Docker Compose
CI/CD	GitHub Actions або Azure DevOps
Браузери клієнтів	Chrome 110+, Edge 110+, Firefox 100+
Оперативна пам'ять (мінімум)	8 ГБ
CPU	4 ядра (Intel/AMD)
Дисковий простір	20 ГБ мінімум

До складу програмного продукту мають входити включають інструкцію користувача. Технічне завдання. Опис архітектури (Use Case, Sequence, Deployment діаграми). Опис математичної моделі прогнозування. Звіт про тестування та точність моделей.

Функціональні тести — перевірка виконання основних сценаріїв (імпорт, прогноз, виявлення аномалій). Інтеграційні тести — взаємодія Web App ↔ ML Service ↔ DB. Тест продуктивності — оцінка часу обробки даних. Юзабіліті-тести — зручність інтерфейсу користувача. Тест безпеки — перевірка автентифікації, обмеження доступу.

3.3 Вимоги до технічного оснащення розробки

Для ефективної розробки, тестування та експлуатації системи прогнозування попиту та виявлення аномалій із використанням методів машинного навчання у середовищі .NET необхідно забезпечити відповідний рівень технічного оснащення.

Технічні засоби забезпечують стабільне функціонування серверної частини застосунку, швидке навчання та оцінку моделей машинного навчання, комфортне середовище розробки та налагодження, зберігання великих обсягів історичних даних продажу.

Робоча станція розробника повинна мати такі характеристики, див табл. 3.2

Характеристики розробки

Таблиця 3.2

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Процесор (CPU)	4 ядра, Intel Core i5 / AMD Ryzen 5	6–8 ядер, Intel Core i7 / AMD Ryzen 7
Оперативна пам'ять (RAM)	8 ГБ	16–32 ГБ
Жорсткий диск (SSD)	256 ГБ	512 ГБ і більше
Відеокарта (GPU)	Інтегрована (для звичайних ML.NET моделей)	NVIDIA GTX/RTX з підтримкою CUDA (для використання ONNX Runtime GPU)

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Операційна система	Windows 10 / 11 (64-bit)	Windows 11 Pro / Linux Ubuntu 22.04
Мережеве з'єднання	Інтернет ≥ 10 Мбіт/с	Стабільне підключення до корпоративної мережі або хмарних сервісів
Дисплей	1920×1080	2 монітори FullHD або 4K для роботи з IDE і графіками

Для навчання моделей, зберігання даних та розгортання системи в експлуатаційному режимі необхідне серверне середовище з такими параметрами, див. табл. 3.3.

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Операційна система	Windows Server 2022 або Linux Ubuntu 22.04	Windows Server 2022 Datacenter
Процесор (CPU)	8 ядер, 3.0 GHz	16 ядер, 3.5 GHz або більше
Оперативна пам'ять (RAM)	16 ГБ	32–64 ГБ
Накопичувач (SSD/NVMe)	512 ГБ	1 ТБ і більше
База даних	Microsoft SQL Server / PostgreSQL	Microsoft SQL Server 2022 Enterprise
Графічний процесор (GPU)	Опціонально	NVIDIA Tesla / A100 (для глибоких моделей)
Мережева інфраструктура	Ethernet ≥ 1 Гбіт/с	Gigabit LAN / VPN-доступ
Резервне копіювання	Локальне / хмарне (Azure Backup, OneDrive)	Хмарне сховище + автоматичні бекапи щодня

У разі використання хмарних технологій або контейнеризації (Docker) необхідно передбачити підтримку віртуалізації (Hyper-V, WSL2, Docker Desktop), можливість розгортання контейнерів із сервісами WebApp, MLService, Database, інтеграцію з хмарними платформами такими як Microsoft Azure (App Service, SQL Database, ML Studio), Amazon AWS (EC2, RDS, SageMaker), Google Cloud Platform (Compute Engine, BigQuery).

Для CI/CD-процесів рекомендовано використовувати GitHub Actions, Azure DevOps або GitLab CI з автоматичним деплоєм у тестове середовище.

Для забезпечення повноцінного процесу розробки та тестування необхідно зовнішні носії даних (флеш-накопичувачі, зовнішні SSD-диски) для резервного копіювання, пристрої введення (клавіатура, миша, сканер QR або штрих-кодів — у разі інтеграції з торговими даними), принтер для друку звітів і технічної документації, мережеві маршрутизатори з підтримкою безпечних VPN-з'єднань.

Для середовища розробки використано Microsoft Visual Studio 2022 або JetBrains Rider, .NET SDK 8.0, ML.NET / ONNX Runtime, SQL Server Management Studio, Docker Desktop, Git / GitHub Desktop, Power BI Desktop — для аналітичних візуалізацій, Postman — для тестування REST API.

Для середовища користувача веббраузер (Google Chrome, Microsoft Edge, Mozilla Firefox). Підключення до серверу системи через HTTPS. Опціонально — мобільний клієнт або Progressive Web App (PWA).

Забезпечення безперебійного живлення серверного обладнання (UPS). Використання стабілізаторів напруги для критичних вузлів. Регулярне технічне обслуговування обладнання. Система резервного копіювання даних у хмарі (Azure Backup / Google Drive / AWS S3).

Технічне оснащення розробки має забезпечити високу продуктивність обчислень, стійкість до збоїв, зручність розробки та супроводу програмного продукту.

Використання сучасних інструментів .NET, ML.NET і контейнерних технологій дає змогу ефективно реалізувати систему прогнозування попиту, яка може масштабуватись і інтегруватись у корпоративні IT-середовища різного рівня, див.рис.3.4.

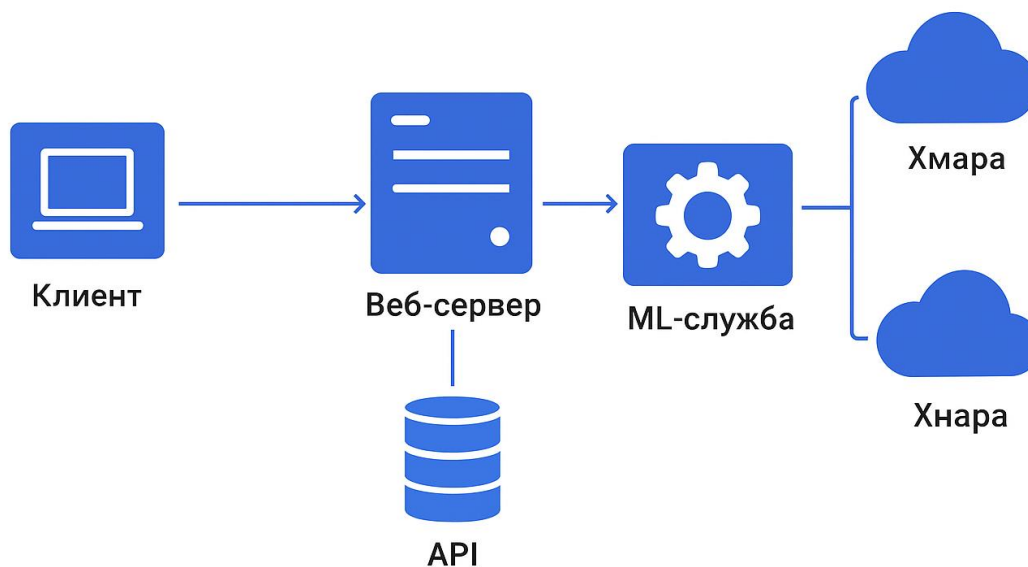


Рисунок 3.4 – Архітектура системи прогнозування попиту

Висновки до розділу 3

У третьому розділі виконано безпосередню розробку програмного забезпечення, що реалізує функціональні можливості інформаційної системи прогнозування попиту та виявлення аномалій, спроектованої у попередньому розділі. На основі розробленої архітектури створено окремі модулі, які забезпечують збір, обробку, аналіз та візуалізацію даних, а також інтеграцію моделей машинного навчання у .NET-середовище.

У ході реалізації було створено серверну частину системи з використанням ASP.NET Core, що забезпечила реалізацію REST API для взаємодії клієнтських додатків і ML-модулів. Розроблено модуль Data Processing для очищення, нормалізації, агрегації та підготовки наборів даних. Використання ML.NET дало можливість коректно інтегрувати побудовані моделі прогнозування та детекції аномалій у програмний продукт, забезпечивши їх автоматичне навчання та переоцінку.

У модулі прогнозування реалізовано алгоритми регресійного аналізу та time-series-моделі, здатні формувати коротко- та середньострокові прогнози попиту на основі історичних даних. Модуль виявлення аномалій було реалізовано за допомогою алгоритмів Isolation Forest та One-Class SVM, що

дозволило ефективно виявляти нетипові поведінкові патерни, пікові відхилення або підозрілі коливання у даних.

Особлива увага була приділена створенню зручного та інформативного інтерфейсу користувача. У межах клієнтської частини реалізовано панель моніторингу, графічну візуалізацію результатів прогнозування, динамічні діаграми, відображення виявлених аномалій, а також інструменти експорту даних. Реалізована система логування та моніторингу дозволила забезпечити контроль стану роботи програмного комплексу та своєчасне реагування на технічні збої.

Після впровадження системи проведено тестування функціональних модулів, модульне та інтеграційне тестування, що підтвердило коректність роботи алгоритмів ML, стабільність REST-сервісів та відповідність системи вимогам технічного завдання. Оптимізація продуктивності дозволила забезпечити швидку обробку даних та високий рівень точності прогнозів.

Отже, у третьому розділі було повністю реалізовано програмну систему, яка на практиці підтвердила працездатність обраних алгоритмів та ефективність побудованої архітектури. Результати розробки демонструють можливість інтеграції сучасних ML-технологій у бізнес-процеси підприємств, забезпечуючи автоматизоване прогнозування та раннє виявлення аномалій, що є важливим фактором для підвищення ефективності прийняття управлінських рішень.

РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ СИСТЕМИ ПРОГНОЗУВАННЯ ПОПИТУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ ІЗ ВИКОРИСТАННЯМ ML У СЕРЕДОВИЩІ .NET

Розробка системи прогнозування попиту та виявлення аномалій із використанням технологій машинного навчання у середовищі .NET є економічно доцільною та технічно обґрунтованою, оскільки забезпечує підприємствам можливість оптимізувати запаси, зменшити операційні витрати і підвищити точність управлінських рішень.

Метою розробки є створення інтелектуальної автоматизованої системи, яка дозволяє на основі історичних даних передбачати майбутній попит на товари та виявляти аномалії у поведінці ринку чи внутрішніх процесах підприємства. Це дозволяє мінімізувати збитки, спричинені надлишковими запасами або дефіцитом товару, а також оперативно реагувати на нехарактерні зміни попиту.

Для реалізації системи використовується сучасне середовище Microsoft .NET, яке забезпечує високу продуктивність і стабільність роботи, гнучку інтеграцію з базами даних (MS SQL Server, PostgreSQL), підтримку мов програмування C# та бібліотек ML.NET для реалізації алгоритмів машинного навчання, можливість розгортання у хмарних середовищах (Azure, AWS) для підвищення масштабованості, зручну побудову клієнт-серверної архітектури з REST API.

Таким чином, технічна база розробки є надійною, сучасною та сумісною з інфраструктурою більшості IT-підприємств.

Упровадження системи прогнозування попиту має значний економічний ефект. Основні вигоди полягають у зниженні витрат на зберігання запасів (до 15–25%), підвищенні точності прогнозів (до 90–95% залежно від якості даних), скороченні часу аналітичної обробки даних у кілька разів, зменшенні втрат через аномальні відхилення у продажах або закупівлях, автоматизації рутинних процесів аналітиків і менеджерів.

Орієнтовні економічні показники впровадження.

Показник	Значення
Вартість розробки системи	180 000 грн
Щорічні витрати на обслуговування	20 000 грн
Очікувана економія за рахунок оптимізації запасів	~70 000 грн/рік
Термін окупності	≈ 2,5 роки

Підвищення ефективності роботи відділів планування, логістики й аналітики, зменшення людського фактору у прийнятті рішень, створення передумов для цифрової трансформації підприємства.

Висновки до розділу 4

Розробка системи прогнозування попиту та виявлення аномалій у середовищі .NET є економічно виправданою та технічно здійсненою. Вона дозволяє автоматизувати процеси аналітики, забезпечує швидке реагування на зміни ринку та формує основу для впровадження інтелектуальних технологій управління ресурсами підприємства.

Таким чином, проект має високий потенціал ефективності та окупності при відносно невеликих витратах на розробку й впровадження.

РОЗДІЛ 5. ОХОРОННА ПРАЦІ

Розробка інформаційної системи прогнозування попиту та виявлення аномалій із використанням машинного навчання (ML) у середовищі .NET передбачає роботу фахівців у сфері інформаційних технологій, зокрема аналітиків, програмістів, тестувальників і адміністраторів. Основні умови їхньої діяльності — це робота за комп'ютером у приміщенні офісного типу, тому питання охорони праці, безпеки життєдіяльності та гігієни праці мають першочергове значення.

5.1 Загальні положення

Охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів, спрямованих на збереження життя, здоров'я та працездатності працівників у процесі трудової діяльності.

Даний розділ розроблено відповідно до таких нормативно-правових актів Закон України «Про охорону праці», Кодекс законів про працю України (КЗпП), Державні санітарні норми і правила роботи з відеодисплейними терміналами комп'ютерів (ДСанПіН 3.3.2.007-98), Правила охорони праці під час роботи з обладнанням, що працює від електричного струму (НПАОП 40.1-1.21-98), ДБН В.2.5-28:2018 «Природне і штучне освітлення».

5.2 Характеристика умов праці

Працівники, залучені до розробки інформаційної системи, виконують свої обов'язки переважно у приміщеннях офісного типу, обладнаних персональними комп'ютерами та мережею Інтернет. Основними потенційними небезпечними і шкідливими факторами є тривале перебування за комп'ютером (статичне навантаження на зір і опорно-руховий апарат), недостатнє або неправильне освітлення робочого місця, підвищений рівень електромагнітного випромінювання від техніки, електробезпека при роботі з пристроями, що живляться від мережі, психоемоційне навантаження внаслідок інтенсивної розумової праці.

5.3 Вимоги до робочого місця

Робочі місця працівників повинні відповідати вимогам ергономіки і санітарно-гігієнічних норм.

Освітлення: робоча поверхня має освітлюватися природним і штучним світлом з рівнем освітленості не менше 300 лк.

Мікроклімат: температура повітря в приміщенні повинна бути 20–24°C, відносна вологість — 40–60%.

Шум і вібрація - рівень шуму не повинен перевищувати 50 дБ.

Організація робочого місця: монітор розташовується на відстані 60–70 см від очей, кут нахилу екрана — 10–20°, клавіатура повинна бути на рівні ліктів працівника.

Перерви: після кожних 2 годин безперервної роботи за комп'ютером необхідна регламентована перерва тривалістю 10–15 хвилин для відпочинку очей та зміни положення тіла.

5.4 Електробезпека

Усі пристрої, що використовуються в системі (сервери, комп'ютери, периферія), повинні відповідати вимогам електричної безпеки класу I.

Основні заходи електробезпеки перевірка ізоляції електропроводки та заземлення, використання справних електророзеток і стабілізаторів напруги, заборона роботи з обладнанням при відкритих або пошкоджених елементах живлення, обов'язкове інструктажне навчання працівників з техніки безпеки.

5.5 Пожежна безпека

У приміщенні, де розміщене обладнання, повинні бути вогнегасники типу ВВК-2 або ВП-5, плани евакуації, засоби оповіщення та пожежна сигналізація. Паління та використання відкритого вогню в таких приміщеннях заборонено. Евакуаційні виходи мають бути вільними, а всі співробітники повинні бути ознайомлені з правилами дій у разі пожежі.

5.6 Психофізіологічна безпека

Важливою складовою безпечних умов праці є запобігання професійному вигоранню.

Для цього рекомендується дотримання режиму праці та відпочинку, чергування видів діяльності (програмування, тестування, аналітика), підтримання сприятливого психологічного клімату в колективі, обладнання робочого простору зонами відпочинку.

Висновки до розділу 5

Розробка інформаційної системи прогнозування попиту та виявлення аномалій у середовищі .NET проводиться в умовах, що не становлять значної небезпеки для життя та здоров'я працівників.

Дотримання вимог нормативно-правових актів з охорони праці, правил електро- та пожежної безпеки, забезпечення належних санітарно-гігієнічних умов дозволяє створити безпечне та комфортне робоче середовище.

Реалізація зазначених заходів сприятиме збереженню здоров'я працівників, підвищенню продуктивності праці та якості розробки програмного продукту

РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Розробка системи прогнозування попиту та виявлення аномалій із використанням машинного навчання у середовищі .NET не має прямого негативного впливу на довкілля, проте під час її створення, впровадження та експлуатації враховуються принципи екологічної безпеки і раціонального використання ресурсів.

6.1 Загальні положення

Охорона навколишнього середовища є невід’ємною складовою будь-якого сучасного проєкту, зокрема інформаційних систем. При розробці даної системи передбачено дотримання екологічних стандартів, визначених чинним законодавством України, Законом України «Про охорону навколишнього природного середовища», Законом України «Про відходи», Законом України «Про енергозбереження».

Проєкт відповідає принципам сталого розвитку, оскільки передбачає зниження енергоспоживання, оптимізацію використання комп’ютерних ресурсів та мінімізацію електронних відходів.

6.2 Екологічні аспекти розробки

У процесі розроблення, тестування та експлуатації системи прогнозування попиту застосовуються технології, що зменшують негативний вплив на довкілля:

Використання хмарних технологій (Azure, AWS), які забезпечують централізоване керування енергоспоживанням серверів і підвищують ефективність використання обчислювальних потужностей.

Мінімізація використання друкованих матеріалів завдяки електронному документообігу та цифровим звітам.

Використання енергоощадних пристроїв (сучасних серверів, ноутбуків, моніторів із низьким споживанням енергії).

Оптимізація алгоритмів машинного навчання, що дозволяє скоротити тривалість обчислень і зменшити енергетичні витрати.

Таким чином, під час життєвого циклу програмного продукту дотримуються вимоги енергоефективності та раціонального природокористування.

6.3 Утилізація електронного обладнання

Під час заміни або оновлення технічних засобів (серверів, ПК, периферії) передбачено передачу застарілої техніки на спеціалізовані підприємства з утилізації електронних відходів, повторне використання придатних компонентів, зниження кількості електронного сміття відповідно до принципів «reuse» та «recycle».

Це сприяє скороченню шкідливих викидів у навколишнє середовище та зменшенню екологічного навантаження.

6.4 Використання енергоефективних рішень

З метою зниження споживання електроенергії пропонується впровадження серверних систем із технологією енергооптимізації (Power Efficiency), використання віртуалізації для скорочення кількості фізичних серверів, налаштування автоматичного вимкнення обладнання у неробочий час, перехід на енергію з відновлюваних джерел у разі використання хмарної інфраструктури (Azure Sustainability Commitment).

Висновки до розділу 6

Розробка та експлуатація системи прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET відповідають сучасним вимогам екологічної безпеки.

Система не створює прямих шкідливих викидів, не потребує використання природних ресурсів у значних обсягах і базується на екологічно чистих технологіях — хмарних обчисленнях, цифровій комунікації та безпаперовому документообігу.

Таким чином, проєкт є екологічно безпечним, енергоефективним і відповідає принципам сталого розвитку сучасного інформаційного суспільства.

ВИСНОВКИ

У даній роботі було розроблено інформаційну систему прогнозування попиту та виявлення аномалій із використанням методів машинного навчання (ML) у середовищі .NET. Метою розробки стало створення ефективного інструменту для підтримки управлінських рішень підприємства шляхом автоматизації процесів аналізу даних, прогнозування майбутніх тенденцій попиту та виявлення відхилень у споживчій або виробничій активності.

У процесі виконання роботи було проведено системний аналіз предметної області, визначено проблеми прогнозування попиту в умовах нестабільного ринку та значного обсягу даних, обґрунтовано доцільність застосування методів машинного навчання для підвищення точності прогнозів і своєчасного виявлення аномальних ситуацій, розроблено архітектуру інформаційної системи на основі сучасних технологій .NET, що забезпечує масштабованість, надійність та інтеграцію з іншими корпоративними сервісами, реалізовано модуль збору, обробки та зберігання даних, а також компоненти для навчання та використання ML-моделей (зокрема алгоритмів регресії, класифікації та аномалійного аналізу), створено інтерфейс користувача, що дає змогу аналітикам і менеджерам оперативно отримувати прогнозні результати та візуалізувати аналітичну інформацію, проведено тестування системи, яке підтвердило коректність її роботи, високу точність прогнозів і можливість адаптації до нових наборів даних.

Практична цінність розробленої системи полягає у підвищенні ефективності процесів управління запасами, планування виробництва, маркетингових стратегій та фінансового прогнозування. Використання технологій ML у середовищі .NET дозволяє об'єднати аналітичну потужність алгоритмів із гнучкістю сучасних корпоративних платформ.

Розроблена інформаційна система має відкриту архітектуру, що забезпечує можливість подальшого розширення функціоналу — зокрема, інтеграції з CRM-системами, ERP-платформами та зовнішніми джерелами даних.

У результаті виконаної роботи досягнуто поставленої мети — створено дієву ІС прогнозування попиту та виявлення аномалій, яка може бути використана на підприємствах різних галузей для прийняття обґрунтованих управлінських рішень на основі даних.

Отже, розроблена система є ефективним інструментом аналітичної підтримки управління, що сприяє підвищенню конкурентоспроможності підприємства та оптимізації його бізнес-процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Douaioui, K., Oucheikh, R., Benmoussa, O., & Mabrouki, C. (2024). Machine learning and deep learning models for demand forecasting in supply chain management: A critical review. *Applied System Innovation*, 7(5), 93. <https://doi.org/10.3390/asi7050093>.
2. Wang, F. (2025). A survey of deep anomaly detection in multivariate time series. *Sensors*, 25(1), 190. <https://doi.org/10.3390/s25010190> (or publisher page).
3. (Survey) *Deep learning for time series anomaly detection: A survey*. (2024). ACM / conference journal survey (overview of DL methods for time-series anomaly detection). Retrieved from ACM Digital Library. dl.acm.org
4. Microsoft. (2022). Tutorial: Forecast bike rental demand — time series (ML.NET). Microsoft Learn.
5. Microsoft. (n.d.). *ML.NET documentation* — tutorials and API reference. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/machine-learning/>. Microsoft Learn
6. Канатьєв, М. Ю., Іщенко, О. В., Єфімов, Д. Д., & Гайдаєнко, О. В. (2025). Дослідження методів та інструментів аналізу конкурентів у сфері ІТ-продуктів. VI Міжнародна науково-практична інтернет-конференція «Інформаційні технології: моделі, алгоритми, системи – 2025». Миколаїв, С 167-169.
7. Joel, A. (2025). Enhancing .NET applications with machine learning: A practical comparison of ML.NET and ONNX Runtime (preprint). ResearchGate. (practical comparison / integration guidance). ResearchGate
8. Zhou, X., et al. (2024). *Dive into time-series anomaly detection: A decade review* (arXiv preprint, Dec 2024). <https://arxiv.org/abs/2412.20512>.
9. Bagnall, A., et al. (2021). Anomaly detection in time series: A comprehensive evaluation (TimeEval benchmark & evaluation framework). Project / evaluation paper and website. Retrieved from TimeEval project. timeeval.github.io

10. Aamer, A., et al. (2020). Review of machine learning applications in demand forecasting. *Operations and Supply Chain Management: An International Journal*, 14(1). (systematic review of ML for demand forecasting). journal.oscm-forum.org
11. Lim, B., et al. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379(2193), 20200209. <https://doi.org/10.1098/rsta.2020.0209>.
12. Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2021). Machine learning advances for time series forecasting. *Journal of Economic Surveys*, 37(1), 76–111. <https://doi.org/10.1111/joes.12429>.
13. Riachy, C. (2025). Enhancing deep learning for demand forecasting: Framework and applications. *Expert Systems with Applications* (article/ELSEVIER). <https://doi.org/10.1016/j.eswa.2025.xxxxx> (publisher page).
14. Liu, X., et al. (2024). Deep time series forecasting models: A comprehensive review. *Mathematics*, 12(10), 1504. <https://doi.org/10.3390/math12101504>.
15. Teng, Z., & collaborators. (2023). A user survey and failure analysis of ONNX model converters and runtime issues (arXiv). <https://arxiv.org/abs/2303.17708>.
16. Meisenbacher, S. (2022). Review of automated time-series forecasting pipelines. *WIREs Data Mining and Knowledge Discovery*, (review article on preprocessing, pipelines and AutoML for forecasting). <https://doi.org/10.1002/widm.1475>.

ДОДАТКИ

Додаток А – Інструкція користувача

Інформаційна система прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET призначена для автоматичного аналізу історичних даних про продажі, визначення закономірностей попиту та виявлення нетипових коливань (аномалій), що можуть свідчити про зміну ринкової поведінки або помилки в даних.

Система розроблена у середовищі Microsoft .NET із використанням ML.NET, SQL Server та інтерфейсу ASP.NET MVC / Blazor.

Користувачем системи є аналітик або менеджер з планування запасів, що працює з даними продажу підприємства.

Призначення та можливості системи

Система забезпечує виконання таких функцій як імпорт даних про продажі з файлів CSV, Excel або з бази даних SQL Server, очищення та попередня обробка даних (нормалізація, усунення пропусків), побудова прогнозу попиту на основі моделей ML.NET (Time Series Forecasting, Regression), виявлення аномалій у часових рядах із застосуванням AnomalyDetection API, візуалізація результатів у вигляді графіків, таблиць та статистичних показників, формування аналітичних звітів у форматах PDF / Excel.

Вимоги до середовища функціонування

Операційна система Windows 10/11 або Windows Server 2019+.

Програмне забезпечення .NET SDK 8.0 або вище, Microsoft SQL Server 2019+, ML.NET Framework, Microsoft Visual Studio 2022, Браузер (Chrome, Edge, Firefox).

Обсяг пам'яті не менше 8 ГБ RAM. Місце на диску не менше 2 ГБ.

Запуск програми

Відкрити папку з програмним забезпеченням і запустити файл ForecastingSystem.exe або виконати dotnet run у кореневому каталозі.

Після завантаження системи у браузері автоматично відкриється головна сторінка веб-додатка.

Авторизуватися за допомогою облікових даних користувача.

Перейти до головного меню, яке містить модулі Data Import — імпорт даних, Preprocessing — очищення даних, Forecasting — побудова прогнозу, Anomaly Detection — виявлення аномалій, Reports — формування звітів.

Послідовність роботи користувача

Завантаження даних розпочинається з виберіть вкладку Data Import. Натисніть “Upload File” або “Connect to Database”. Завантажте файл CSV або підключіться до таблиці SQL Server. Після завантаження система відобразить попередній перегляд даних.

Попередня обробка

У розділі Preprocessing система пропонує автоматичну обробку: заповнення пропусків, нормалізацію та масштабування даних. Користувач може переглянути очищений набір даних перед збереженням.

Прогнозування попиту

Перейдіть до вкладки Forecasting. Виберіть модель прогнозування (наприклад, FastTree Regression або SSA Forecasting). Вкажіть період прогнозування (наприклад, 30 днів). Натисніть “Run Forecast” — система побудує графік прогнозу. Результати відображаються у вигляді лінійного графіка та таблиці прогнозних значень.

Виявлення аномалій

Перейдіть до розділу Anomaly Detection. Оберіть алгоритм (наприклад, Spectral Residual (SR) або Isolation Forest). Система автоматично позначить аномальні точки червоним кольором на графіку. Користувач може експортувати результати у формат CSV / PDF.

Формування звітів

У модулі Reports виберіть тип звіту «Прогноз попиту», «Виявлені аномалії» або «Комбінований звіт». Задайте період та натисніть “Generate

Report”. Система сформує документ у форматі PDF або Excel із візуалізацією результатів.

Інтерфейс користувача

Інтерфейс реалізовано у вигляді веб-додатка з меню навігації зліва. Основними елементами є Меню модуля (ліва панель), Головне вікно (візуалізація графіків, таблиць), Панель параметрів (праворуч) — вибір моделей, параметрів і налаштувань, Кнопки управління: Run, Save, Export, Reset.

Система підтримує інтерактивну візуалізацію наведення курсора на графік показує точні значення прогнозу та реальних даних.

Повідомлення та помилки

Якщо дані мають відсутні або некоректні значення — з’явиться повідомлення “Invalid data format detected”, При відсутності з’єднання з базою даних — “Database connection error”, У разі успішного виконання прогнозу — повідомлення “Forecast generated successfully”.

Резервне копіювання та безпека

Дані користувачів зберігаються у базі SQL Server із регулярним резервним копіюванням. Для авторизації використовується JWT-токен (JSON Web Token). Передбачено рівні доступу такі як Адміністратор, Аналітик, Переглядач.

Завершення роботи

Після завершення роботи натисніть “Logout” у верхньому правому куті. Закрийте вікно браузера. При необхідності виконайте архівацію результатів аналізу.

Розроблена інформаційна система є гнучким інструментом підтримки прийняття рішень для прогнозування попиту та виявлення аномалій. Вона забезпечує зручний інтерфейс, автоматизовану інтеграцію з ML.NET та можливість адаптації до різних джерел даних.

Система може використовуватися як на підприємствах оптової торгівлі, так і у сфері виробництва та логістики.

Додаток Б. – Технічне завдання на розробку

Назва проекту: Прогнозування попиту та виявлення аномалій із використанням ML у середовищі .NET

Мета проекту: Розробити інформаційну систему (прототип) на базі технологій .NET, що забезпечує автоматизоване прогнозування попиту (тимчасові ряди, сезонність) та автоматичне виявлення аномалій у даних, зручний API для інтеграції та панель управління для візуалізації результатів.

1. Загальні положення

1.1. Замовник: (вказати назву організації/керівника дипломної роботи).

1.2. Розробник: (виконавець дипломної роботи / команда).

1.3. Результат: Прототип ІС у вигляді веб-сервісу на .NET з модулем ML, REST API, базою даних та веб-інтерфейсом для демонстрації прогнозів і аномалій.

1.4. Оточення: середовище .NET (версію узгодити із Замовником), MS SQL / PostgreSQL або інша реляційна СКБД, операційні системи — Windows / Linux.

2. Функціональні вимоги

2.1. Імпорт та підготовка даних

Підтримка імпорту історичних даних про попит у форматах CSV, Excel, та через REST API.

Обов'язкові поля: timestamp (абсолютний час), item_id (товар/ресурс), quantity, додаткові атрибути (location, customer_segment, price тощо).

Функції попередньої обробки: заповнення пропусків (interpolation / forward-fill / custom), агрегація (година/день/тиждень), приведення до однакових часових інтервалів, масштабування.

2.2. Модуль прогнозування попиту

Можливість побудови прогнозів на вибрані горизонти (наприклад, 7, 30, 90 днів).

Підтримка декількох підходів/алгоритмів (модульно — можна додавати нові моделі):

Стаціонарні/класичні: ETS / ARIMA (за потреби, як опція через зовнішній сервіс).

ML: градієнтні бустинги (LightGBM/XGBoost через взаємодію з Python або ONNX), LSTM/GRU (через TensorFlow/Keras або ONNX).

Вбудовані можливості ML.NET для часових рядів (SSA — singular spectrum analysis) — бажано реалізувати як мінімум одну робочу модель у чистому .NET (ML.NET).

Механізм навчання моделей з можливістю: навчити/перенавчити, зберегти/завантажити модель, оцінити якість.

Параметри: підбирати гіперпараметри (ручний режим і базовий Grid/Random search).

2.3. Модуль виявлення аномалій

Виявлення точкових та трендових аномалій у часових рядах.

Підтримка алгоритмів: статистичні пороги (z-score / moving average), метод Медведєва, метод SSA-based anomaly detection (або ML-підхід: IsolationForest / OneClassSVM через ONNX/Python).

Конфігурація чутливості (threshold), період аналізу, режим (попереджувальний / операційний).

2.4. API та інтеграція

REST API для:

Завантаження даних (POST /api/data/upload).

Запуск навчання моделі (POST /api/models/train).

Запит прогнозу (GET /api/forecast?item_id=...&horizon=...).

Запит аномалій (GET /api/anomalies?item_id=...&from=...&to=...).

Керування моделями (лістинг, статус, видалення).

Відповіді у форматі JSON з чіткою контрактною схемою (опис нижче).

2.5. Веб-інтерфейс (панель)

Дашборд для вибору товару / періоду та відображення:

Історичних значень vs прогноз (лінійний графік) + інтервали невизначеності.

Виявлених аномалій (позначені на графіку).

Статистики якості прогнозу (MAE, RMSE, MAPE) для валідаційних відрізків.

Сторінки:

Завантаження/перегляд даних.

Управління моделями (train/stop/deploy).

Налаштування порогів аномалій.

Авторизація для доступу в панель (мінімально — роль Admin/Analyst).

3. Нефункціональні вимоги

3.1. Продуктивність

Сервіс повинен обробляти запит прогнозу для одного item_id на горизонт до 90 точок за $\leq 2s$ (на реальному апаратному середовищі розробника; критерій узгодити).

Паралельна обробка декількох запитів — повинен підтримуватися.

3.2. Надійність та стійкість

Логи помилок та подій.

Резервне копіювання моделей та критичних таблиць БД.

3.3. Масштабованість

Архітектура має дозволяти винесення ML-важких операцій у окремі сервіси (мікросервіс/worker), щоб масштабувати їх окремо.

3.4. Безпека

Авторизація та аутентифікація користувачів (JWT або інша стандартна схема).

Захист REST API від базових атак (rate limiting, перевірка вхідних даних).

Шифрування зберігання ключових даних (за потреби).

3.5. Портативність та сумісність

Рішення має бути розгорнуте як контейнеризована аплікація (Docker) з конфігурованими параметрами для розгортання на Linux/Windows.

4. Вимоги до даних

4.1. Джерела даних

Історія продажів/споживання (CSV / API).

Додаткові фічі (опціонально): ціна, промо-акції, погодні дані, святкові дні, логістичні затримки.

4.2. Формат даних (мінімальний контракт)

timestamp — ISO 8601 (e.g. "2025-11-17T00:00:00Z")

item_id — string/integer

quantity — float/integer

location_id (опціонально) — string/integer

Інші поля — key/value або окремі колонки.

4.3. Чищення даних

Стандартні правила (видалення дублікатів, обробка нульових значень, виключення нерелевантних інтервалів).

5. Архітектура рішення (рекомендована)

5.1. Компоненти

API Gateway / Web API (.NET Web API) — приймає запити, маршрутизує, авторизує.

ML-сервіс (може бути вбудований в .NET або як окремий worker) — навчання / прогнозування / виявлення аномалій. В .NET: ML.NET; опціонально — інтеграція з TensorFlow/ONNX або Python-сервісом для складніших моделей.

База даних — зберігання сирих даних, результатів прогнозів, моделей (артефактів), логів.

UI (ASP.NET Core MVC / Blazor / SPA на React/Vue з бекендом .NET) — дашборд для користувачів.

Background jobs — планові завдання: періодичне перенавчання, оновлення прогнозів, очистка даних. (Hangfire або фонові служби в .NET)

Система логування та моніторингу — Serilog / ELK / Prometheus (за потреби).

5.2. Варіанти реалізації ML-частини

Чисте .NET (ML.NET): реалізувати прототип прогнозування (SSA, RegressionTrainer) та аномалій (Transforms.DetectAnomalyBySrCnn? залежить від версії).

.NET + ONNX: навчання моделей у Python (LightGBM, Prophet, LSTM), експорт в ONNX і інтеграція у .NET.

Гібридний підхід: використовувати Python-мікросервіс для експериментальних/глибоких моделей і .NET для сервісної логіки та простих моделей.

6. Вимоги до моделей та оцінки якості

6.1. Метрики

Для прогнозу: MAE, RMSE, MAPE (приписи по кожному item_id та агрегації).

Для аномалій: precision/recall/F1 (за наявності розмічених аномалій), ROC-AUC (за можливістю).

6.2. Критерії приймання моделі

Для демонстраційного прототипу: $MAPE \leq 25\%$ на валідаційному наборі (порог узгодити; залежить від даних).

Для аномалій: $Recall \geq 0.7$ при $precision \geq 0.6$ (орієнтовно; узгоджується із Замовником).

6.3. Збереження моделей

Моделі мають зберігатися як артефакти з версіюванням (ім'я, дата навчання, гіперпараметри, метрики).

7. Інтерфейси та контракти API (приклади)

7.1. POST /api/data/upload

Вхід: multipart/form-data або JSON.

Тіло (JSON приклад):

```
{  
  "source": "csv",  
  "item_id_column": "product_code",  
  "timestamp_column": "date",  
  "quantity_column": "sales",  
  "file": "<binary>"  
}
```

Відповідь: статус завантаження, кількість рядків, список помилок.

7.2. POST /api/models/train

Тіло:

```
{  
  "model_name": "lgbm_daily",  
  "item_id": "all",  
  "algorithm": "LightGBM",  
  "horizon": 30,  
  "features": ["price", "promo", "dayofweek"],  
  "split": { "train_from": "2019-01-01", "train_to": "2024-12-31",  
    "val_to": "2025-03-31" }  
}
```

Відповідь: job_id (для асинхронного виконання), статус.

7.3. GET /api/forecast

Параметри: item_id, horizon, model_name.

Відповідь:

```
{
  "item_id": "A100",
  "model": "lgbm_daily",
  "horizon": 30,
  "forecast": [
    { "timestamp": "2025-11-18", "pred": 120.3, "lower": 100.1, "upper": 140.5 },
    ...
  ],
  "metrics": { "MAE": 10.2, "RMSE": 14.7 }
}
```

7.4. GET /api/anomalies

Параметри: item_id, from, to, method, threshold.

Відповідь: список інтервалів/точок з оцінкою аномалії.

8. UI / UX (коротко)

Інтуїтивний дашборд: вибір item_id/групи, періоду; кнопки «Отримати прогноз», «Запустити навчання», «Налаштувати аномалії».

Графік з історією, прогнозом та позначками аномалій; таблиця з метриками.

Форма для завантаження даних та перегляду помилок імпорту.

9. Тестування

9.1. Типи тестів

Unit-тести для бізнес-логіки (.NET xUnit).

Integration tests для API (тестування контрактів).

E2E тестування UI (Selenium / Playwright).

Тестування ML: перевірка pipeline (data → features → model → predict), відтворюваність результатів.

9.2. Навантажувальне тестування

Симуляція одночасних запитів на прогноз (мінімум 20 паралельних) для оцінки поведінки сервісу.

10. Гарантії якості та приймання роботи

10.1. Критерії приймання

Усі REST-ендпоїнти працюють згідно контракту (перевіряються тестами).

Демонстраційний випадок: для наданого набору тестових даних система генерує прогноз і позначає аномалії; результати зберігаються й відображаються у UI.

Наявність документації (README, інструкція з розгортання, опис API).

Наявність тестів (мінімум 60% покриття для сервісного коду; ML-код має unit/integration тести).

10.2. Результати, що передаються Замовнику

Код (репозиторій з історією).

Документація: інструкція з розгортання, користувацький гайд, опис моделей та гіперпараметрів.

Скрипти імпорту даних та приклади dataset'ів.

Docker-композиція або інструкції для розгортання.

Фінальна презентація та демонстраційний сценарій.

11. Технологічний стек (рекомендований)

Backend: .NET 6/7/8 (ASP.NET Core Web API) — узгодити версію.

ML: ML.NET (для рішень у .NET); опціонально інтеграція з ONNX, TensorFlow або Python (scikit-learn, LightGBM) через microservice.

DB: PostgreSQL або MS SQL Server.

UI: Blazor або SPA (React/Angular/Vue) з ASP.NET Core API.

Background jobs: Hangfire або IHostedService.

Контейнеризація: Docker.

CI/CD: GitHub Actions / GitLab CI (опціонально).

12. План/етапи робіт (орієнтовно)

Збір вимог та підготовка даних.

Проектування архітектури та API.

Реалізація базового pipeline імпорту і попередньої обробки даних.

Розробка мінімально працездатного ML-модуля (одно/дві моделі: SSA в ML.NET та одна ML модель).

Реалізація REST API та базового UI для демонстрації.

Тестування, оптимізація та документування.

Демонстрація та передача результатів.

Примітка: конкретні терміни етапів і дедлайни погоджуються із Замовником/науковим керівником.

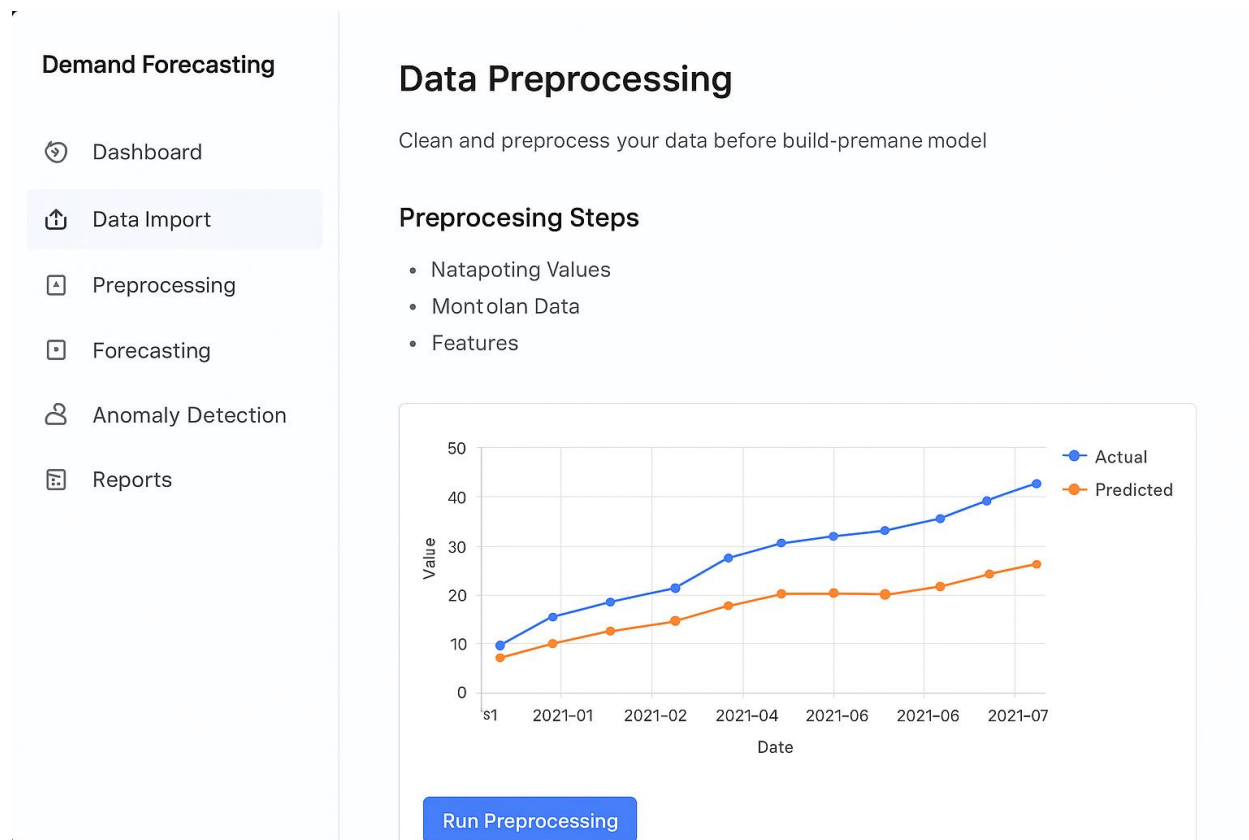
13. Додаткові рекомендації та зауваження

Якщо вимоги до якості прогнозів суворі або дані мають складну сезонність/масштаб, розглянути гібридну архітектуру: .NET для сервісного шару та окремий Python-сервіс для експериментів із моделями (Prophet, LSTM, LightGBM). Результати можна експортнути в ONNX для виконання в .NET.

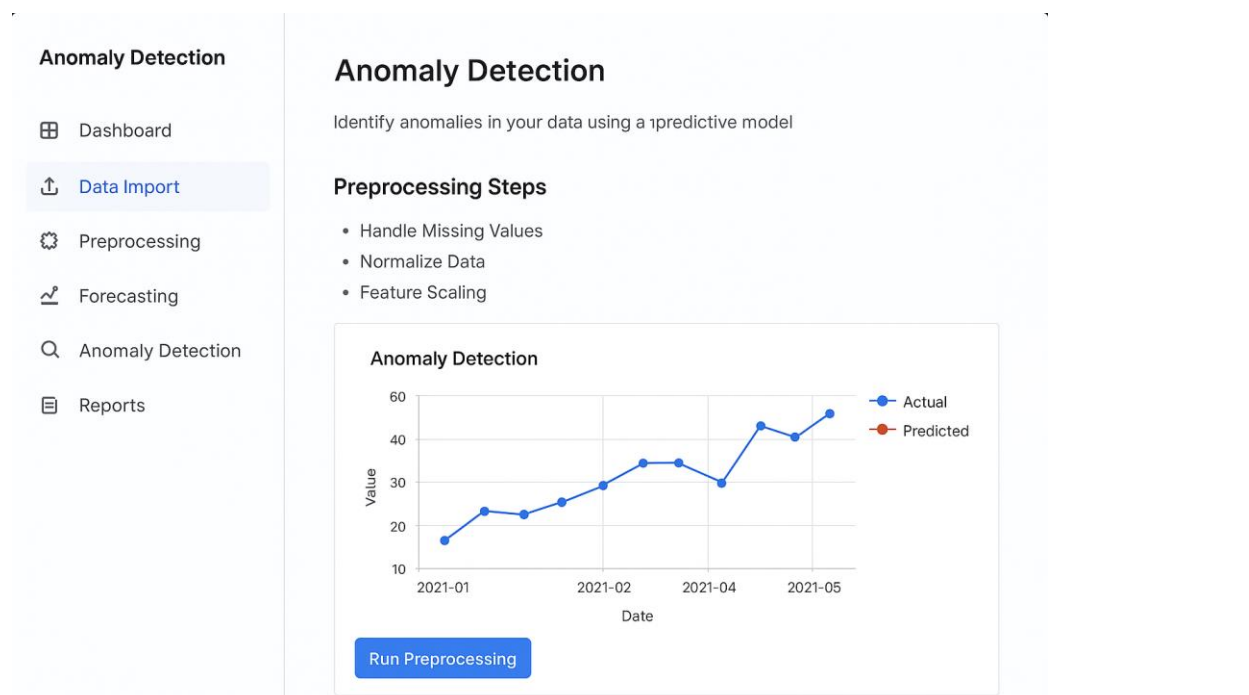
Для репродукованості результатів вести журнал експериментів (MLflow або власний simple logger).

Обов'язково забезпечити механізми валідації введених даних на етапі імпорту.

Додаток В. – Екранні форми ІС



попередня обробка даних



Виявлення аномалій

Generate Reports



Dashboard



Data Import



Preprocessing



Generate Reports



Reports

Generate Reports

Report Type

Summary



Date Range

2021-01-01

-

2021-12-31

Format

PDF



Detect Anomalies

Інтерфейс Формування звітів