КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітньо-наукова програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Розробка технології прогнозування фінансового бюджету методами Data Science»

Студентки 2-го курсу групи ІАВ-21
Катерина НЕМЧЕНКО
(прізвище, ім'я, по батькові)

(підпис студента)

Науковий керівник:
Доктор технічних наук, доцент
(науковий ступінь, вчене звання)
Хлевна Юлія Леонідівна
(прізвище, ім'я, по батькові)

(дата) (підпис)

Попередній захист:

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри
технологій управління

(підпис) (прізвище, ініціали) (дата)

Київ – 2025

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
Професор Морозов В.В.
_____
«___» _____ 2025 року

ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент: Катерина НЕМЧЕНКО

Група: ІАВ-21

1. **Тема кваліфікаційної роботи магістра**: «Розробка технології прогнозування фінансового бюджету методами Data Science».
Затверджена протоколом засідання кафедри ТУ №___ від _____ року
2. **Строк подання студентом готової роботи** – «19» травня 2025 р.
3. **Цільова установка та вихідні дані до роботи**: Дослідження зосереджене на розробці й впровадженні комплексної технології короткострокового (1–3 місяці) прогнозування особистого фінансового бюджету на основі транзакційних даних із застосуванням різноманітних методів Data Science. У роботі обґрунтовано вибір і послідовність застосування наївних (Naive, Seasonal-Naive), експоненційного згладжування (ETS), моделі SARIMAX із екзогенними змінними (курси валют, індекс споживчих цін, свята), бізнес-орієнтованого Prophet, нелінійного XGBoost із лаг-ознаками та спеціалізованих методів Croston/TSB для інтермітентних витрат. Розроблено правило-диспетчер для автоматичного підбору моделі за видом витрат, впроваджено expanding-window cross-validation для оцінки точності та інтеграцію прогнозів у Power BI. *Вихідні дані до роботи*: CSV-виписки транзакцій з Monobank (містять дати, суми, MCC-коди, описи операцій); CSV-виписки транзакцій з ПриватБанку (дата, сума, опис); Довідник MCC-кодів та словник відповідності до категорій витрат; Курс валют UAH → USD із відкритого API НБУ; Календар державних і персональних свят, дані про вихідні (бібліотека holidays для Python); Індекси споживчих цін (CPI) Держстату України; Нормативні документи й методичні матеріали з фінансового планування домогосподарств; Огляд наукових джерел щодо короткострокового прогнозування фінансових часових рядів.
4. **Зміст роботи**: У кваліфікаційній роботі проведено аналітичний огляд предметної області та характеристику вихідних даних із транзакцій банківських карток; досліджено теоретичні основи ключових методів прогнозування фінансових часових рядів (наївні, ETS, SARIMAX, Prophet, XGBoost, Croston/TSB); реалізовано підготовку даних, включаючи очищення, агрегацію за категоріями та місяцями, інженерію

екзогенних ознак; виконано побудову й розширювальну крос-валідацію моделей із порівняльним аналізом точності за метриками MAPE та wMAPE; розроблено та впроваджено механізм автоматичного відбору моделей (диспетчер за категоріями витрат); створено програмний модуль на Python для навчання й інференсу моделей; виконано економічне обґрунтування доцільності застосування технології в особистому й корпоративному фінансовому плануванні.

5. **Перелік графічного матеріалу**: Робота містить 18 рисунків (діаграми місячних витрат, STL-декомпозиція, heatmap кореляцій, схема expanding-window CV, архітектура диспетчера моделей, UML-діаграма, скріншоти дашборду Power BI тощо); 5 таблиць (опис вихідних даних, характеристики екзогенних змінних, структури об’єднаних датасетів, класифікація категорій та ін.); 24 формули — нумерація від (1) до (24) охоплює MAPE, sMAPE, MASE, параметричні рівняння SARIMAX, ETS та інші теоретичні вирази, що використовуються у розділах 2-3.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1	Вибір теми дипломної роботи	3	10.12.24	10.12.24
2	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	27.12.24	27.12.24
3	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	08.01.25	07.01.25
4	Складання розгорнутого плану кваліфікаційної роботи	5	18.01.25	18.01.25
5	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	5	19.01.25 - 20.01.25	20.01.25
6	Підготовка розділу 1 «АНАЛІЗ ПІДХОДІВ, МЕТОДІВ ТА АЛГОРИТМІВ DATA SCIENCE ДЛЯ ФОРМУВАННЯ Й ПЛАНУВАННЯ БЮДЖЕТІВ»	10	12.02.25	13.02.25
7	Підготовка розділу 2 «ТЕОРЕТИЧНІ ОСНОВИ МОДЕЛЕЙ ПРОГНОЗУВАННЯ»	14	08.03.25	08.03.25
8	Підготовка розділу 3 «ПІДГОТОВКА ДАНИХ І РЕАЛІЗАЦІЯ АЛГОРИТМІВ ДЛЯ ПРОГНОЗУВАННЯ КОРОТКОСТРОКОВИХ ВИТРАТ»	14	20.03.25	20.03.25
9	Підготовка розділу 4 «ІНТЕГРАЦІЯ, ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ТА ПОДАЛЬШИЙ РОЗВИТОК РІШЕННЯ ДЛЯ ПРОГНОЗУВАННЯ БЮДЖЕТУ»	13	15.04.25	15.04.25
10	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	15	25.04.25	25.04.25
11	Передача кваліфікаційної роботи науковому керівникові	2	01.05.25	01.05.25
12	Передача кваліфікаційної роботи рецензенту для рецензування	2	04.05.25	04.05.25
13	Попередній захист кваліфікаційної роботи	5	13.05.25	13.05.25

Дата видачі завдання «01» грудня 2024 р.

Керівник роботи: д.т.н., доц. Хлевна Юлія Леонідівна _____

(підпис)

Завдання прийняла до виконання студентка групи ІАВ-21

Немченко Катерина Юріївна

(прізвище, ім’я, по батькові)

ЗМІСТ

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра технологій управління
Спеціальність 122 - Комп'ютерні науки,
освітня програма "Інформаційна аналітика та впливи"

Дипломна робота магістра Немченко Катерини Юріївни.

Тема роботи: «Розробка технології прогнозування фінансового бюджету методами Data Science».

Мета дипломної роботи магістра – розробити й експериментально обґрунтувати технологію короткострокового (1–3 місяці) прогнозування особистого фінансового бюджету, що поєднує класичні статистичні й сучасні машинні методи та забезпечує інтеграцію результатів у аналітичні системи бізнес-інтелекту.

Об'єкт дослідження – процес формування та виконання особистого фінансового бюджету на основі транзакційних даних банківських рахунків.

Предмет дослідження – методи, моделі та програмні засоби Data Science для короткострокового прогнозування бюджетних показників за умов обмежених і нерегулярних часових рядів, а також правила їх автоматичного вибору за категоріями витрат.

Наукова новизна роботи полягає у розробці комплексної технології, що: поєднує наївні, ETS, SARIMAX із екзогенними змінними (курси валют, календар свят), Prophet, XGBoost із лаг-ознаками та спеціалізований Croston/TSB для інтермітентних витрат; упроваджує розширювальне перехресне валідування (expanding-window CV) для коротких рядів і доводить статистично значуще зниження MAPE порівняно з базовими підходами; пропонує правило-диспетчер вибору моделі за типом витрат, що підвищує точність й інтерпретованість прогнозів; демонструє практичну інтеграцію модуля прогнозування у Power BI безпосередньо на рівні візуалізацій.

У роботі проаналізовано існуючі підходи до застосування статистичних та машинних методів у прогнозуванні фінансових бюджетів. Запропонована нова методика їх комбінування й адаптації до коротких особистих часових рядів, обґрунтовано доцільність її впровадження, наведено рекомендації щодо практичної імплементації у корпоративні та персональні аналітичні системи.

Дипломна робота складається зі вступу, чотирьох розділів, висновків і списку використаних джерел. Загальний обсяг – _ сторінок; список літератури містить _ найменування на _ сторінках.

Ключові слова: фінансовий бюджет, прогнозування, Data Science, SARIMAX, Prophet, XGBoost, Croston, короткострокові часові ряди.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

AIC (Akaike Information Criterion) – інформаційний критерій Акаіке для відбору моделей.

AICc (Corrected Akaike Information Criterion) – скорегований AIC з додатковим штрафом за кількість параметрів у малих вибірках.

API (Application Programming Interface) – програмний інтерфейс взаємодії між системами (наприклад, REST-API НБУ для курсу USD↔UAH).

ARIMA (AutoRegressive Integrated Moving Average) – авторегресійна інтегрована модель ковзної середньої для опису часових рядів.

BI (Business Intelligence) – рішення для інтерактивного аналізу даних та дашбордів.

BIC (Bayesian Information Criterion) – байєсівський інформаційний критерій, який жорсткіше карає складність моделі, ніж AIC.

CI/CD (Continuous Integration / Continuous Delivery) – практики безперервної інтеграції коду та автоматизованого розгортання сервісів.

CRISP-DM (Cross-Industry Standard Process for Data Mining) – галузева методологія шести етапів проєктів Data Science.

CRPS (Continuous Ranked Probability Score) – метрика для оцінювання повного прогнозного розподілу.

Croston – спеціалізований метод прогнозування переривчастих (інтермітентних) рядів.

CSV (Comma-Separated Values) – табличний формат файлу з комами як роздільниками.

CPI (Consumer Price Index) – індекс споживчих цін, що вимірює зміну вартості кошика товарів і послуг.

EDA (Exploratory Data Analysis) – розвідувальний аналіз даних на початковому етапі.

ETL (Extract-Transform-Load) – процес витягу, трансформації та завантаження даних у сховище.

ERP (Enterprise Resource Planning) – інтегровані системи управління ресурсами підприємства.

ES-RNN (Exponential Smoothing Recurrent Neural Network) – гібрид Holt–Winters та рекурентної нейронної мережі, переможець M4.

ETS (Error-Trend-Seasonality) – експоненційне згладжування Холта-Вінтерса (рівень, тренд, сезонність).

KPI (Key Performance Indicator) – ключовий показник ефективності бізнес-цілей.

LLM (Large Language Model) – великі мовні моделі на кшталт GPT, застосовувані для класифікації транзакцій.

LightGBM (Light Gradient Boosting Machine) – швидка реалізація градієнтного бустингу дерев від Microsoft.

MAE (Mean Absolute Error) – середня абсолютна помилка прогнозу.

MASE (Mean Absolute Scaled Error) – MAE, нормоване на середню однокрокову зміну ряду.

MAPE (Mean Absolute Percentage Error) – середня абсолютна відносна помилка, виражена в %.

MCC (Merchant Category Code) – чотиризначний код категорії торговця у карткових транзакціях.

ML (Machine Learning) – машинне навчання для автоматичного виявлення закономірностей.

Naïve / Seasonal Naïve – бенчмаркові методи: копіюють останнє або сезонне значення ряду.

PSI (Population Stability Index) – індекс стабільності розподілу даних для моніторингу дрейфу моделей.

Prophet – адитивна байєсівська модель із piece-wise трендом і гармонічною сезонністю.

RMSE (Root Mean Squared Error) – корінь із середньоквадратичної помилки прогнозу.

SBA (Syntetos–Boylan Adjustment) – поправка до класичного Croston для зменшення зміщення прогнозу.

SARIMA (Seasonal ARIMA) – ARIMA з сезонними компонентами.

SARIMAX (Seasonal ARIMA with eXogenous factors) – SARIMA, що включає зовнішні регресори.

sMAPE (Symmetric MAPE) – симетрична версія MAPE, рівноважна до недопрогнозів і перепрогнозів.

STL (Seasonal-Trend decomposition using Loess) – декомпозиція ряду на тренд, сезонність і залишки.

TBATS (Trigonometric Box-Cox ARMA Trend Seasonal) – модель для складних сезонних патернів з Box-Cox перетворенням.

Theta-метод – статистичний підхід, розклад ряду на дві θ-лінії; переможець M3.

TSB (Teunter-Syntetos-Babai) – удосконалена версія Croston з окремим згладжуванням імовірностей.

wMAPE (weighted MAPE) – зважена MAPE, що враховує різні обсяги категорій.

wRMSE (Weighted RMSE) – RMSE з можливістю призначення різної ваги спостереженням.

XGBoost (Extreme Gradient Boosting) – градієнтний бустинг дерев із підтримкою лагових ознак.

ВСТУП

Раціональне планування фінансів – незалежно від того, йдеться про сімейний чи організаційний бюджет – є базовою умовою стійкості. Проте інфляція, коливання валютних курсів, сезонні піки споживання енергоносіїв, разові витрати на подарунки або премії, а також нерегулярні платежі на кшталт квартальних страховок чи щорічних підписок суттєво ускладнюють прогнозування.

Проблема загострюється, коли історичних даних ще обмаль: людина лише почала отримувати стабільну зарплату й фіксувати витрати, стартап веде облік перший рік, або новий підрозділ компанії не накопичив довгої фінансової статистики. Такі часові ряди короткі й містять пропуски, через що класичні методи (наприклад SARIMA) швидко втрачають точність. Водночас популярні фінансові застосунки здебільшого покладаються на спрощені середні значення й майже не враховують зовнішніх чинників. Тим часом екзогенні впливи – насамперед курси валют і календар свят – систематично зміщують структуру витрат і мають бути інтегровані в модель.

За цих умов на часі розробка технології Data Science, здатної працювати з обмеженою історією транзакцій, зважати на зовнішні фактори й формувати надійні короткострокові прогнози (1–3 місяці). Помилкові оцінки витрат призводять до касових розривів, вимушених кредитів або урізання критично важливих статей бюджету; натомість точний прогноз дає змогу завчасно резервувати кошти на сезонні піки, оптимізувати підписки й уникати штрафів за прострочені платежі.

Мета дослідження – розробити й експериментально обґрунтувати технологію прогнозування фінансового бюджету, що поєднує класичні та сучасні методи Data Science, коректно обробляє короткі й нерівномірні часові ряди та автоматично підбирає модель з урахуванням характеру кожної категорії витрат і впливу екзогенних чинників, забезпечуючи підвищену точність короткострокових прогнозів для різних сценаріїв бюджетування.

Запропонована технологія повинна не лише підвищити середню точність прогнозу на горизонті 1–3 місяці, а й гарантувати відтворюваність результатів і можливість інтеграції в популярні фінансові застосунки та системи управлінського обліку. Це створить практичну основу для свідомого фінансового планування за мінімальної історії даних і високої невизначеності зовнішнього середовища.

.

РОЗДІЛ 1. АНАЛІЗ ПІДХОДІВ, МЕТОДІВ ТА АЛГОРИТМІВ DATA SCIENCE ДЛЯ ФОРМУВАННЯ Й ПЛАНУВАННЯ БЮДЖЕТІВ

1.1 Аналіз сфери бюджетування та роль прогнозів у ній

Бюджет як інструмент фінансового управління виник разом із першими централізованими державами й донині залишається ключовою ланкою у процесі планування ресурсів [1]. У публічному секторі бюджетування визначає пріоритети соціальних програм, інвестиції в інфраструктуру та можливості фіскальної стабілізації. Для бізнесу воно виконує роль «фінансового компаса»: допомагає розподіляти капітал між підрозділами, контролювати витрати, оцінювати рентабельність і приймати стратегічні рішення щодо дивідендної політики чи розширення ринків. Значущість бюджетування посилилася в умовах постпандемічної турбулентності, коли компанії й уряди одночасно зіштовхнулися зі стрибками інфляції, розривами ланцюгів постачання й коливаннями валютних курсів [2] [3]. За цих обставин короткострокова достовірність бюджетних прогнозів стала критичним фактором сталого розвитку. Саме на такі «непрецедентні умови» якраз і спрямовують гнучкі ML-моделі бюджетного прогнозу [4].

Широке впровадження інформаційних систем управління (ERP, EPM-платформи, BI-дашборди) трансформувало традиційний процес складання бюджетів. Автоматизовані консолідації даних, сценарне моделювання, what-if-аналіз і контроль відхилень (variance analysis) поступово витіснили громіздкі електронні таблиці.

У корпоративному середовищі Data Science-алгоритми вже відіграють вагому роль: від сезонно-авторегресійних моделей для прогнозу продажів до градієнтного бустингу в задачах оцінки попиту на сировину [4] [5]. Консалтингові огляди показують, що ML допомагає скоротити ручну роботу, підвищуючи точність фінансових прогнозів і даючи змогу оперативно переорієнтовувати ресурси [6] [4].

Своє відображення це знаходить і на макрорівні: центральні банки застосовують ансамблі статистичних і машинних моделей для вдосконалення макроекономічних прогнозів, зокрема інфляції та ВВП [7] [8].

Прикладне підтвердження бачимо й на макрорівні: у консультативному звіті МВФ для Центрального банку Йорданії описано перехід від «чистого» ARIMA до ансамблевого фреймворку із динамічним вибором та комбінуванням моделей, що підвищило точність прогнозу готівки в обігу [9].

Муніципалітети, у свою чергу, використовують подібні підходи для планування енергоспоживання та транспорту.

Емпіричні дослідження, зокрема конкурс M4 (100 тис. рядів, 61 метод), довели, що гібрид експоненційного згладжування з RNN (ES-RNN) систематично випереджає ізольовані статистичні чи суто ML-моделі, підкріплюючи тезу про необхідність ансамблів [10].

Тенденція до персоналізації фінтех-сервісів винесла бюджетування на рівень домогосподарств, але у більшості випадків користувачі й далі покладаються на прості «ручні» схеми. Найпоширеніші з них – «конверти» (cash-stuffing) [11]; правило 50/30/20, популяризоване Е. Воррен у All Your Worth [12]; нуль-базове бюджетування, яке позиціонують як альтернативу інкрементному підходу [13]; та саморобні Excel-шаблони. Усі ці методи дисциплінують витрати [14], проте не враховують сезонності, інфляційних трендів і валютних ризиків, переносячи на користувача весь когнітивний тягар. Саме на ці обмеження вказує й The Guardian, фіксуючи одночасний бум ZBB-шаблонів і потребу у «цифрових помічниках» [15].

Фінтех уже частково закриває цю прогалину: Mint автоматично тегує транзакції [16], Monobank показує детальний зріз витрат і кешбек-категорій, Revolut Insights пропонує дашборди й ліміти без корекції на CPI [17], а YNAB зосереджується на ретроспективних «Spending Trends», залишаючи forward-прогноз користувачеві [18] [19] [20]. У підсумку нинішні сервіси автоматизують збір і візуалізацію даних, але не дають прозорого, категорійного прогнозу майбутніх витрат – розрив, який і формує дослідницьку проблему цієї роботи.

Блок «планування майбутніх витрат» зазвичай лишається напівручним: користувачеві пропонують самостійно ввести ліміти для кожної категорії, спираючись лише на інтуїцію або швидкий перегляд середніх витрат за минулі періоди. Саме цей розрив між традиційними ручними методами та частково «розумними» інструментами визначає дослідницьку проблему, яку розв’язує дана робота.

З позиції Data Science задача прогнозування особистого бюджету має низку специфічних труднощів. По-перше, часові ряди витрат по окремих категоріях зазвичай короткі: навіть чотири роки щомісячних даних – це лише сорок із лишком спостережень, чого недостатньо для більшості глибинних моделей [21]. По-друге, ряди нерідко містять пропуски й нульові спостереження, характерні для нерегулярних платежів (страховка, медичні послуги). По-третє, вагомий вплив екзогенних чинників – індексу споживчих цін, офіційного курсу НБУ, календаря свят – ускладнює прості методи на кшталт ковзних середніх.

Експериментальне дослідження коротких рядів (35 серій) засвідчило, що за умов високого шуму найкращим прогнозом може бути навіть спрощене сезонне рухоме середнє, тоді як складні ансамблі переобтяжуються [22]. Ці особливості роблять особисте бюджетування перспективним майданчиком для комбінування статистичних моделей (Theta, Holt-Winters) зі «легкими» ML-алгоритмами та економічними регресорами.

Практичний імпульс до цієї роботи народився з власного досвіду: щомісяця, плануючи витрати у мобільному банку, доводилося з нуля встановлювати ліміти приблизно для п’ятнадцяти категорій. Підписки на онлайн-сервіси, страхові платежі й сезонні рахунки регулярно відхилялися від плану, спричиняючи відхилення до 20 % від бажаного бюджету. Очевидним стало, що навіть грубий автоматичний прогноз – своєрідний baseline – суттєво зменшив би ризик помилок і зекономив час. Водночас рішення мало б залишатися прозорим і простим в інтеграції з існуючим BI-дашбордом, аби його можна було оперативно оновлювати і порівнювати з фактичними витратами.

Таким чином, у межах дослідження пропонується розробити та оцінити технологію короткострокового прогнозування бюджетних показників, що поєднує інтерпретовані статистичні моделі з легковаговими алгоритмами Data Science і враховує зовнішні економічні фактори. Очікується, що отримані результати маютимуть не тільки практичне значення для особистого фінансового планування, а й можуть бути масштабовані на малі бізнес-підрозділи чи нечисленні проєктні бюджети, де даних так само бракує для «важких» моделей, але потрібна прийнятна точність і швидке впровадження..

1.2 Методологія проєкту

Для системної розробки технології прогнозування бюджету обрано **CRISP-DM (Cross-Industry Standard Process for Data Mining)** – найпоширеніший у Data Science цикл із шести фаз, що поєднує чітку послідовність дій із можливістю повертатися на попередні етапи у разі появи нових даних чи змін бізнес-вимог. Ця методологія залишається найбільш усталеним стандартом організації робіт як в академічному середовищі, так і в індустріальних застосуваннях, особливо у фінансовому секторі [23] [24]. Її структура забезпечує чіткі контрольні точки, ітеративність та гнучкість, що особливо критично при роботі з короткими, щомісячними рядами особистих витрат. Нижче розглянуто, як кожна фаза моделі CRISP-DM інтерпретується теоретично та реалізується на практиці у межах даного проєкту.

Business Understanding (розуміння бізнес-вимог) передбачає формалізацію завдання мовою бізнес-метрик. Теоретично це означає пошук відповіді на запитання «що саме вважається успіхом?». У нашому випадку успіх визначено чіткими KPI: сумарний *MAPE* прогнозу не більший за 10 %, а помилка для кожної з категорій – не більша за 20 % на горизонті 1–3 місяці; додатковим обмеженням є потреба у інтеграції з дашбордом Power BI та переривчасті витрати.

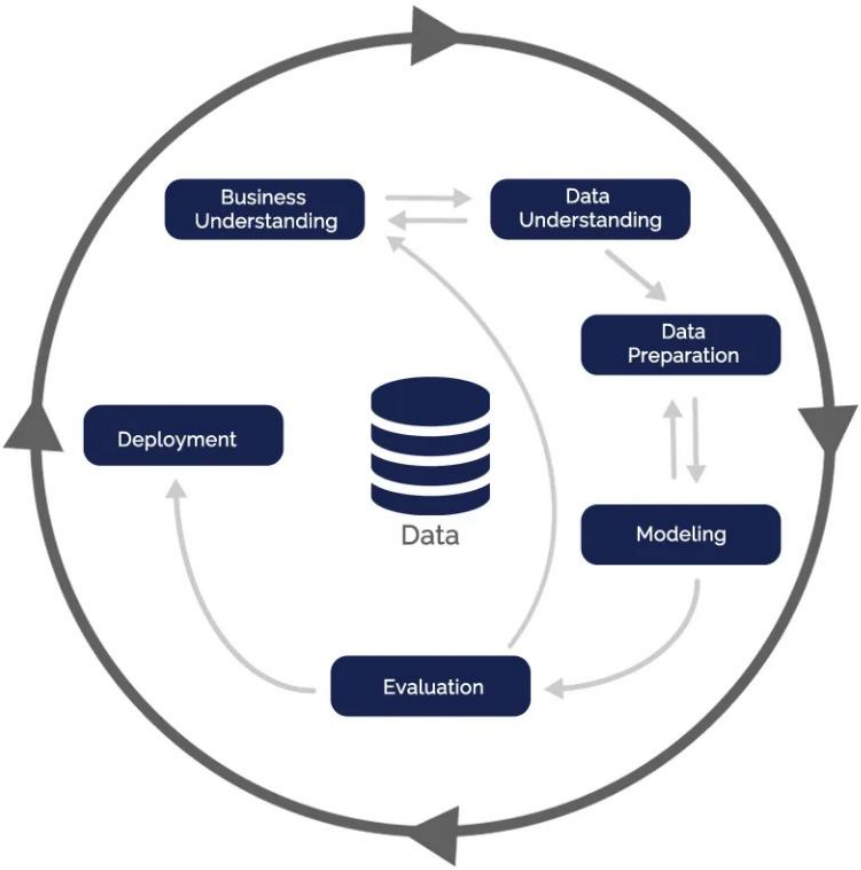

Рисунок 1.1 – Схематична ілюстрація методології CRISP-DM [25].

Data Understanding (розуміння даних) у CRISP-DM покликаний дати всебічне уявлення про джерела, структуру й якість даних. У проєкті аналізується ~4 000 транзакцій (Monobank та Privat24) за 01-2022 – 03-2025 рр. і виявляються характерні закономірності: зимова сезонність комунальних платежів, піки витрат перед святами, а також можлива кореляція витрат з курсом USD/UAH. До набору одразу додаються зовнішні чинники – офіційний курс та індекс споживчих цін, календар державних свят та список особистих свят.

Data Preparation (підготовка даних) у теорії охоплює очищення, інтеграцію й перетворення даних, необхідні, щоби «сирі» транзакції стали придатними для моделювання. Практично це реалізовано скриптом…. Усі суми вже номіновані у гривні, видалено дублікати й повернення, транзакції зіставлено з 15 категоріями за MCC-кодами й словником ключових слів, а далі агреговано у місячні ряди $X_{c,t}$; екзогенні змінні $Z_{k,t}$ синхронізовано з тими самими часовими мітками.

Modeling (моделювання) згідно з CRISP-DM означає відбір методів, побудову й налаштування моделей. Тут використано ієрархію від простого до складного: Seasonal Naïve як бенчмарк → SARIMA/SARIMAX → TBATS та Prophet для мультисезонності → XGBoost із лаг-фічами як нелінійний підхід; для переривчастих категорій тестується Croston. Перебір гіперпараметрів виконується у схемі **expanding-window cross-validation** (первинне вікно 24 місяці, крок 1 місяць, горизонт h = 1-3) – підхід рекомендовано для фінансових часових рядів як такий, що найкраще відтворює реальний режим експлуатації [26].

Evaluation (оцінка) завершує технічне коло CRISP-DM і перевіряє, наскільки модель відповідає бізнес-цілям. У проєкті моделі оцінюються за *MAPE, wMAPE* і *MAE* на останніх шести місяцях; додатково контролюється стабільність прогнозу: різниця між фактичними й прогнозними витратами не повинна виходити за межі 1,5 σ історичного відхилення. Якщо пороги KPI порушуються, ітерація повертається до фази Modeling.

Deployment у класичному розумінні CRISP-DM – це перенесення моделі в реальне середовище й організація її циклу життя. Спочатку прогноз щомісяця генерується Python-скриптом і записується у CSV, який автоматично підхоплює Power BI; далі roadmap передбачає контейнеризацію у хмарні сервіси для повного CI/CD. Щоб контролювати деградацію, щомісяця обчислюється **Population Stability Index (PSI)**; якщо PSI > 0,25 – ініціюється автоматичне перенавчання, оскільки такий поріг трактують як суттєвий drift, особливо у фінансових застосунках [27].

Таким чином, **CRISP-DM забезпечує як теоретичну коректність процесу, так і практичну керованість**: кожна фаза – від постановки задачі до впровадження – супроводжується конкретними артефактами (KPI-документ, EDA-ноутбук, скрипт підготовки, MLflow-запис, метрики CV, Docker-образ), що робить результати відтворюваними, верифікованими та придатними до масштабування на нові джерела або додаткові категорії витрат. Завдяки своїй гнучкій структурі методологія дозволяє швидко реагувати на зміни: додавання

нових даних або екзогенних факторів (наприклад, погодних умов чи змін податкової політики) не порушує логіки побудови моделі, а лише повертає проєкт до відповідної фази циклу. Це особливо важливо при роботі з короткими особистими часовими рядами, де навіть кілька нових місяців спостережень можуть істотно змінити оптимальну модель або вимоги до неї.

1.3 Огляд моделей та алгоритмів бюджетного прогнозування

Мета цього підрозділу – проаналізувати спектр доступних методів прогнозу та окреслити, які з них здатні працювати на коротких (24–48 точок) категорійних часових рядах особистих витрат із потенційними екзогенними факторами (CPI, курс, календар свят).

У класичних статистичних підходах **Naïve** і **Seasonal Naïve** традиційно використаються як базові межі точності: конкурси M-серії показали, що навіть у великій вибірці серій простий «copy-last» залишається гідним орієнтиром для порівняння складніших моделей [28]. На рівні згладжування **ковзне середнє** послаблює випадкові коливання, але не враховує ні тренд, ні сезонність; натомість **експоненційне згладжування Holt–Winters (ETS)** додає обидві ці компоненти, зберігаючи низьку кількість параметрів і стабільність на вибірках у два-три сезони. Практичні посібники з прогнозування (Hyndman & Athanasopoulos) зазначають, що саме ETS зазвичай випереджає ARIMA за коротких історій завдяки меншій варіативності оцінок [29].

ARIMA / SARIMA / SARIMAX, сформульовані в класичній «Time Series Analysis: Forecasting and Control» (Box, Jenkins, Reinsel, Ljung), залишаються золотим стандартом у макрофінансових застосуваннях, даючи можливість моделювати автокореляцію й інтегрувати зовнішні регресори в SARIMAX. Однак процедура Box–Jenkins на ≤ 40 точках часто призводить до переобрання, що потребує жорстких обмежень на порядки та фіксації сезонних лагів [30]. Для даних із більш складною багаторівневою сезонністю актуальною є модель **TBATS**, що комбінує Box-Cox-трансформацію, ARMA-шум і тригонометричні

регресори. Попри гнучкість, автори методу вказують мінімальну бажану довжину серії ≈ 50 спостережень, тож при наших обмеженнях TBATS доцільніше лишити поза експериментом [31].

У сфері машинного навчання помітне поширення отримав **Prophet** (Taylor & Letham) – адитивна модель із piece-wise трендом, гармонійними сезонними складовими та можливістю вбудувати свята або інфляцію. Її створено саме для “коротких бізнес-рядів”, вона стійка до пропусків і дозволяє швидке добирання точок зламу [32]. Підхід **XGBoost із лаг-ознаками** продемонстрував високу точність у конкурсі M5 на 42 840 роздрібних серіях завдяки здатності моделювати нелінійні зв’язки й обробляти нерегулярність [33]. Разом із тим огляд Makridakis et al. (2018) показав, що на вибірках середнього розміру класичні статистичні методи часто залишаються точнішими й менш ресурсоємними, а ML-алгоритми випереджають їх лише за великої кількості рядів або ознак [34].

Серед новітніх глибоких архітектур слід згадати **ES-RNN**, переможця конкурсу M4: гібрид експоненційного згладжування й рекурентної мережі зміг поєднати інтерпретовану тренд-сезонну основу з нелінійним «хвостом» [35]. Проте ті самі автори підкреслюють, що перевага гібридних або чисто нейронних моделей виявляється лише на сотнях або тисячах серій, тоді як для одиничного короткого ряду ризик перенавчання високий. **Theta-метод**, навпаки, зберігає простоту: декомпонує серію на дві “θ-лінії” й дав найкращу точність у M3, що робить його корисним легким орієнтиром між Naïve та ETS [36].

Особливість особистого бюджету – наявність категорій із рідкісними, але значними платежами (страхування, ремонт). Для таких переривчастих рядів класичні ETS або ARIMA зводяться до нульового прогнозу; натомість **Croston** окремо оцінює середній розмір і середній інтервал між транзакціями, тоді як **Teunter–Syntetos–Babai (TSB)** додатково згладжує ймовірність настання події, знижуючи зміщення первісного методу [37].

Зважаючи на обмеження щодо довжини рядів та прагнення до прозорості, для подальших експериментів обрано такі класи моделей: Naïve і Seasonal Naïve

як базові бенчмарки; ETS (Holt–Winters) та SARIMAX для статистичного моделювання з можливістю врахування екзогенних регресорів; Prophet для адитивного розкладу тренду й сезонності; XGBoost із лаг-фічами для виявлення нелінійних залежностей; а також TSB для переривчастих витрат із Croston як допоміжним еталоном. Такий пул дозволяє порівняти прості статистичні, сучасні ML та спеціалізовані алгоритми без надмірної складності та додержання прозорості інтерпретації.

1.4 Опис і аналіз даних

Первинною й методично найціннішою базою для дослідження є транзакції, які протягом кількох років фіксувалися у мобільному застосунку **Saldo**. Саме у Saldo вручну сформовано й багаторазово апробовано остаточний перелік із більш ніж п’ятнадцяти логічно відокремлених категорій витрат, тому саме ця структура вважається «еталонною» для всіх подальших перетворень. Первинний задум передбачав прямий експорт уже категоризованих даних із Saldo, проте такий функціонал у застосунку відсутній, а ручне копіювання непродуктивне. Через це було вирішено працювати з оригінальними банківськими виписками, повторно приводячи їх до логіки Saldo.

Першим джерелом стали **CSV-виписки Monobank**. Файл містить дату й час операції, суму в UAH, окремо суму в оригінальній валюті, чотиризначний MCC-код, текстовий опис і технічні поля, зокрема залишок після транзакції. Практика показала, що MCC-класифікація часом помилкова: цифрові підписки фігурують як «Інше», а покупки проїзних – як «Транспорт». Отже MCC слугує не остаточним тегом, а лише попередньою підказкою.

Другим джерелом є **PDF-виписки PrivatBank**. Дані довелося спершу перетворити у інший формат, виокремлюючи дату, суму, валюту й опис, після чого перетворювати в табличний вигляд. MCC-поле тут відсутнє, тому категорія визначається за ключовими словами в описі або узгоджувати з аналогічними записами у Saldo.

Таблиця 1.1 – **Структура виписки від Monobank**

№	Назва поля	Опис
1	Дата і час операції	Момент проведення транзакції; відображає точну дату та час списання або зарахування коштів на картку.
2	Деталі операції	Текстовий опис контрагента та типу операції (назва торгової точки, сервісу, банкомата, переказу тощо).
3	MCC	Merchant Category Code – чотиризначний код, що відносить контрагента до певної категорії торговельної чи сервісної діяльності (харчі, транспорт, готелі тощо).
4	Сума у валюті картки (UAH)	Фінальна сума операції у валюті карткового рахунку (гривні), розрахована банком із урахуванням курсу на момент проведення.
5	Сума у валюті операції	Початкова сума транзакції у тій валюті, якою здійснювалась операція (наприклад, USD, EUR або UAH); зі знаком “–” для списання та “+” для зарахування.
6	Валюта	Код валюти оригінальної операції (скорочений ISO-код: UAH, USD, EUR тощо).
7	Курс	Обмінний курс банку, за яким сума в іноземній валюті перераховується у валюту картки; порожнє для операцій у гривні.
8	Сума комісій (UAH)	Загальна сума банківських комісій за проведення операції, виражена в гривнях.
9	Сума кешбеку (UAH)	Нарахований за операцію кешбек (повернення частини витрачених коштів), у гривнях; 0,00 – якщо кешбек не застосовується.
10	Залишок після операції	Баланс карткового рахунку відразу після проведення цієї транзакції (у гривнях).

	A	B	C	D	E	F	G	H	I	J
1	Дата і час операції	Деталі операції	MCC	Сума в валюті картки (UAH)	Сума в валюті операції	Валюта	Курс	Сума комісій (UAH)	Сума кешбеку (UAH)	Залишок після операції
2	29.04.2025 15:13:53	Кулиничі	5462	-110	-110	UAH	—	—	0.82	8644.01
3	29.04.2025 12:30:47	Кулиничі	5462	-78	-78	UAH	—	—	0.58	8754.01
4	28.04.2025 15:28:33	Аптека оптових цін	5912	-285.8	-285.8	UAH	—	—	—	8832.01
5	27.04.2025 23:58:06	Spotify	5815	-209.6	-4.99	USD	42.004	—	—	9117.81
6	27.04.2025 20:42:10	Скасування. Bolt	4121	80	80	UAH	—	—	-0.6	9327.41
7	27.04.2025 20:41:20	Bolt	4121	-110	-110	UAH	—	—	0.82	9247.41
8	27.04.2025 20:21:30	Bolt	4121	-80	-80	UAH	—	—	0.6	9357.41
9	27.04.2025 19:34:18	Сільпо	5411	-185.99	-185.99	UAH	—	—	1.39	9437.41
10	27.04.2025 18:54:34	Magazin produktv Moloko v	5499	-239.15	-239.15	UAH	—	—	1.79	9623.4
11	27.04.2025 18:40:08	Тітка Клара	5814	-45	-45	UAH	—	—	—	9862.55
12	27.04.2025 18:31:41	Тітка Клара	5814	-328.5	-328.5	UAH	—	—	—	9907.55
13	27.04.2025 15:46:59	Укрзалізниця	4112	-59.95	-59.95	UAH	—	—	—	10236.05
14	27.04.2025 13:34:22	КСВ KRUASANNI	5812	-164.5	-164.5	UAH	—	—	—	10296
15	27.04.2025 01:45:06	Скасування. Укрзалізниця	4112	460.5	460.5	UAH	—	—	—	10460.5
16	26.04.2025 19:51:17	Поповнення «На підтримку ЗСУ»	4829	-1283.69	-1283.69	UAH	—	—	—	10000

Рисунок 1.2 – Приклад виписки з рахунку Monobank.

Таблиця 1.2 – **Структура виписки від PrivatBank**

№	Назва поля	Опис
1	Дата операції	Дата та час проведення транзакції у форматі ДД.ММ.РРРР ГГ:ХХ; відображає момент списання або зарахування коштів.
2	Рахунок	Маскований номер карткового рахунку (останні 4 цифри); вказує, з якого рахунку списано або на який зараховано кошти.
3	Деталі операції	Опис транзакції – назва торговельної точки чи сервісу, адреса, тип операції (покупка, повернення, переказ тощо).
4	Сума у валюті операції	Первісна сума транзакції та валюта, у якій вона відбулася (UAH); зі знаком “–” для списання, “+” для зарахування.
5	Сума у валюті картки	Еквівалент суми операції в гривні, перерахований банком за внутрішнім курсом.
6	Сума комісій	Нарахована банком комісія за проведення операції (у гривнях).
7	Сума знижок	Нарахований кешбек або інші бонуси/знижки за операцію (в гривнях); 0,00 – якщо знижка не застосовувалася.
8	Залишок після операції	Баланс карткового рахунку одразу після виконання операції (у гривнях).

Дата операції	Рахунок	Деталі операції	Сума у валюті операції	Сума у валюті картки	Сума комісій	Сума знижок	Залишок після операції
04.03.2022 18:54	414960******5189	Продукти: Супермаркет Сильпо, Львів, вул.Шевченка, буд.60	-162,74 UAH	-162,74	0,00	0,00	16 954,77
04.03.2022 10:23	414960******5189	Універмаг: Свій Маркет, Львів, вул.Шевченка, буд.45/47	-17,49 UAH	-17,49	0,00	0,00	17 117,51
04.03.2022 09:50	414960******5189	Ресторан: LVIVSKI KRUASANY, LVIV	-105,00 UAH	-105,00	0,00	0,00	17 135,00
04.03.2022 04:55	414960******5189	Переказ в свою «Скарбничку» 26**99. Округлення залишку по картці до 10 UAH.	-3,03 UAH	-3,03	0,00	0,00	17 240,00
03.03.2022 17:57	414960******5189	Готель: Готель Hostel constellation 89, Львів, вул.Шевченка, буд.21	-2 100,00 UAH	-2 100,00	0,00	0,00	17 243,03
03.03.2022 13:23	414960******5189	Продукти: Близенько, Львів, м.Львів, вул.Шевченка,буд.111а	-46,97 UAH	-46,97	0,00	0,00	19 343,03
03.03.2022 06:24	414960******5189	Продукти: magazyn Target Market, m.Lviv	-60,00 UAH	-60,00	0,00	0,00	19 390,00
02.03.2022 03:50	414960******5189	Переказ в свою «Скарбничку» 26**99. Округлення залишку по картці до 10 UAH.	-9,00 UAH	-9,00	0,00	0,00	19 450,00
01.03.2022 18:14	414960******5189	Телеком послуги: SWEET.TV1, Kyiv	-1,00 UAH	-1,00	0,00	0,00	19 459,00

Рисунок 1.3 – Приклад виписки з рахунку PrivatBank.

Загалом опрацьовано 5 325 транзакцій за період із січня 2020 по березень 2025 року. На першому етапі записи обох банків об’єднані в єдину таблицю, зайві службові операції (перекази між власними рахунками, повернення товарів, списання бонусів) вилучені, а всі суми приведено до гривні за курсом НБУ на дату операції. Далі транзакції зіставлено з еталонними категоріями Saldo:

частково автоматично – за MCC-кодами [38] й словником ключових слів, а частково вручну для спірних випадків, коли опис надто узагальнений.

Для моделювання дані агреговано за принципом «місяць × категорія». До кожного місяця приєднано зовнішні регресори: курс USD/UAH на кінець місяця, бінарний індикатор наявності державних та сімейних свят та індекс споживчих цін. Пропуски витрат заповнено нулями, адже нуль у цьому контексті означає відсутність покупки, а не втрату даних.

Первинний аналіз даних об'єднаних даних показує, що активні витрати почалися з 2022 року (на це вплинули фактори зміни способу життя та доходу) (рис. 1.4). Отже є сенс будувати обробляти дані та будувати моделі починаючи з 2022 року.

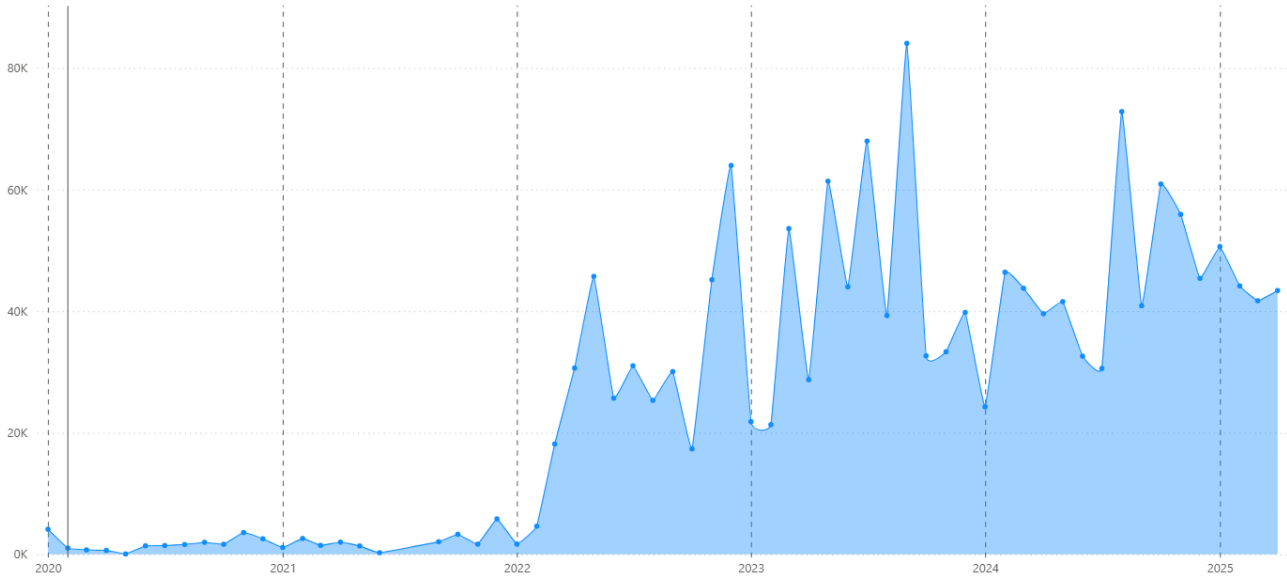

Рисунок 1.4 – Щомісячні витрати починаючи з 2020 року (Power BI)

Також можемо проаналізувати статистику витрат по 10 найбільших категоріях (рис 1.5) та наявність витрат по всіх категоріях помісячно (рис 1.6).

Аналіз транзакцій, зафіксованих із липня 2020 р. до квітня 2025 р., показав виразну багаторівневу структуру поведінки витрат – як у часовому розрізі, так і між категоріями.

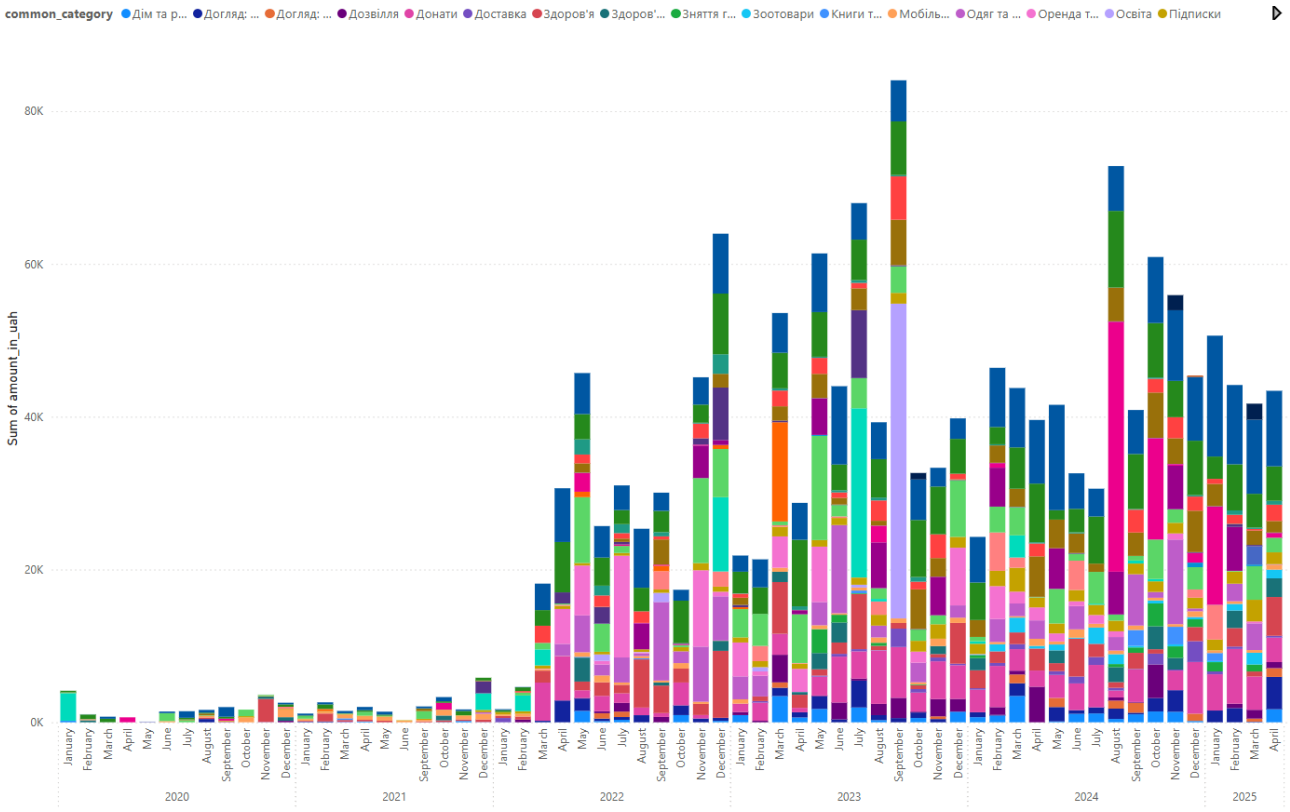

Рисунок 1.5 – Розподіл витрат між категоріями починаючи з 2020 року (Power BI)

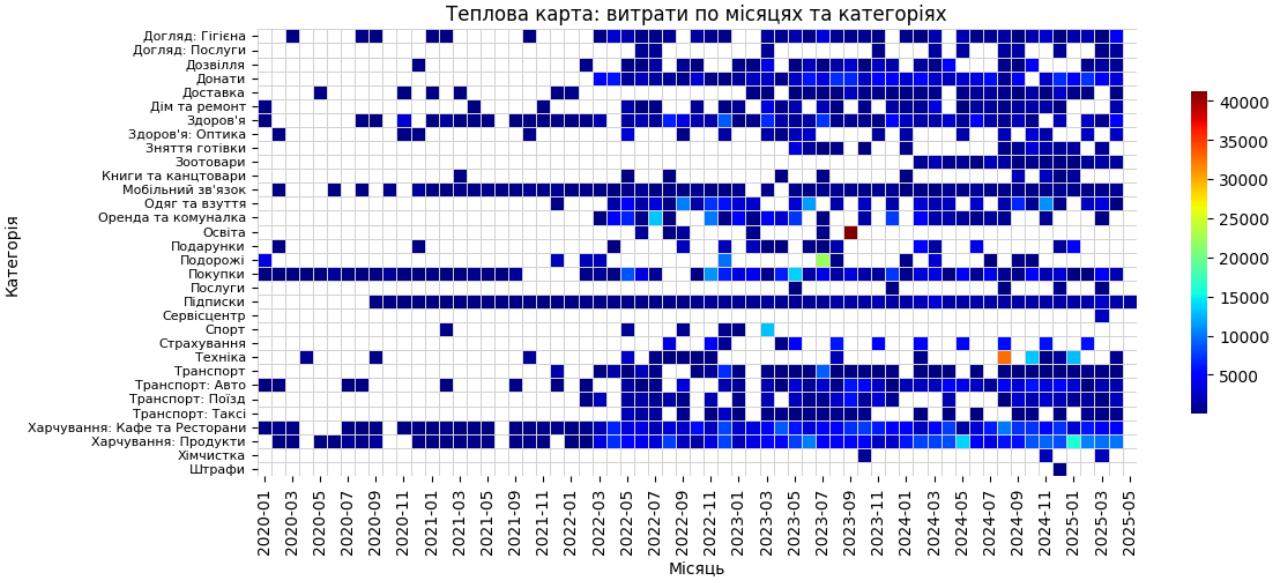

Рисунок 1.6 – Наявність витрат щомісяця та їх величина по кожній з категорій починаючи з 2020 року (matplotlib)

Початкові місяці демонстрували порівняно невеликий обсяг – у середньому близько п’яти тисяч гривень на місяць. У березні-квітні 2022 р. крива

різко змінилася: сумарні витрати зросли майже у більш ніж три рази. Стрибок збігся у часі з вимушеним переїздом та одноразовими закупівлями першої необхідності на початку повномасштабного вторгнення, що засвідчує чутливість домогосподарського бюджету до зовнішніх шоків. Після цього витрати стабілізувались на суттєво вищому рівні, але зберігали хвилеподібний характер: друга помітна серія піків простежується наприкінці 2023 р. і пояснюється великими витратами на подорожі та оплату за університет – саме тоді найвиразніше зростають підкатегорії «Подорожі» та «Освіта».

Якщо вилучити перекази з категорії заощаджень, що маскують реальну споживчу динаміку, ландшафт основних статей витрат змінюється. Найбільший внесок забезпечують дві харчові підкатегорії («Продукти» та «Кафе та ресторан») – сумарно близько чверті всіх витрат за період.

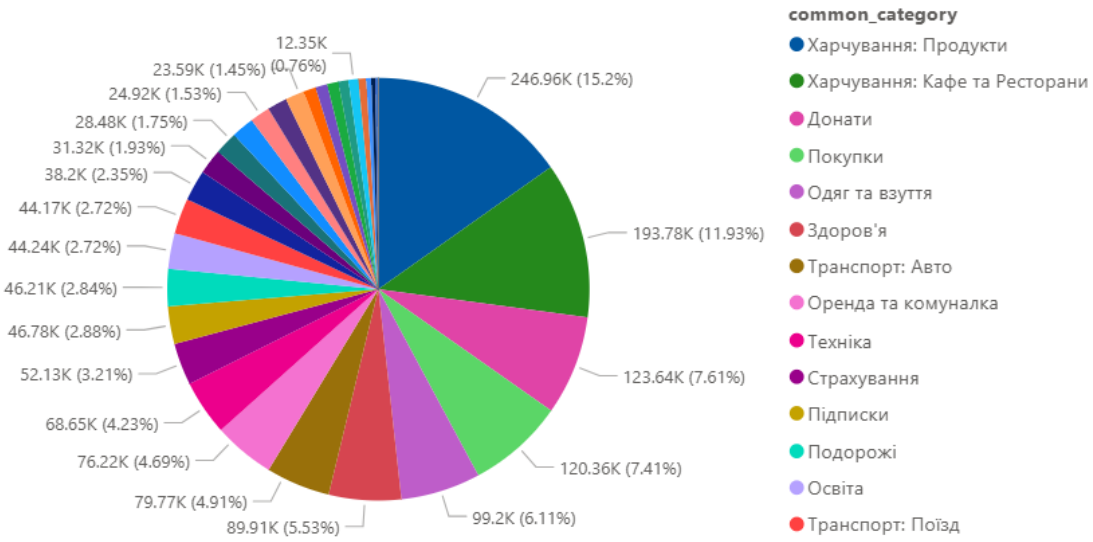

Рисунок 1.7 – Розподіл витрат між категоріями (Power BI)

Подальший розподіл виявляє групи з різною внутрішньою логікою. «Комунальні платежі» мають чітку зимову сезонність; «Подарунки» концентрують платежі у лютому (родинні дати) та грудні; витрати на «Продукти» традиційно спадають у влітку, коли частину харчів покриває власний урожай. Навпаки, «Техніка», «Медичні послуги» та «Подорожі» проявляють себе як спорадичні серії: довгі відрізки нульових значень перериваються одиничними, але великими платежами.

Теплова карта, побудована за матрицею «місяць × категорія», підтверджує ці спостереження. Смуги для щоденних потреб (харчування, транспорт, комунальні) пофарбовані рівномірно і безперервно, тоді як спорадичні категорії вирізняються окремими «спалахами» кольору. Візуальна різниця у щільності та насиченості клітинок дає підстави диференціювати підходи до моделювання. Для регулярних рядів доцільно застосовувати класичні сезонно-авторегресивні моделі з екзогенними регресорами (SARIMAX), які здатні урахувати коливання курсу USD, державні свята й календарні індикатори. Натомість для розріджених послідовностей необхідні методи, спеціально розроблені для інтермітентних даних, – Croston або його модифікація TSB; у диспетчері моделей поріг вибору цих алгоритмів визначено часткою нульових місяців (≈ 40 %).

Сукупність графіків також окреслює вимоги до попередньої трансформації. Рідкісні, але екстремальні витрати породжують довгий хвіст розподілу, тому перед навчанням моделей варто застосовувати логарифмування, вінзоризацію або сценарійне сімплінг-перевірку, аби послабити вплив аномальних точок.

Таким чином, емпіричний аналіз формує дві групи вимог. По-перше, модельний пул має бути гібридним: SARIMAX/SARIMA та Prophet – для стабільних сезонних рядів; Croston/TSB – для інтермітентних; XGBoost з лагами – як нелінійна альтернатива для категорій із перехресними ефектами. По-друге, у підготовчому етапі необхідно реалізувати автоматичне виявлення та згладжування шокових витрат, а також зберегти інформацію про календарні й валютні змінні, що пояснюють частину дисперсії.

Узагальнюючи, сформовано цілісний датасет, що поєднує достовірність ручної категоризації Saldo з повнотою банківських виписок і доповнений зовнішніми макропоказниками. Саме на цій основі вибудовуються подальші експерименти з прогнозування.

1.4 Постановка задачі

Нехай $X_{c,t}$ – матриця місячних витрат по $c = 1 \dots 20$ категоріях у період із січня 2022 по березень 2025 року, а $Z_{k,t}$ – матриця екзогенних факторів (курс USD/UAH, CPI, бінарні ознаки свят). Потрібно побудувати функцію $f : (X_{\cdot,\le t}, Z_{\cdot,\le t}) \to \hat{y}_{c,t+h}$ таким чином, щоб на горизонтах $h = 1,2,3$ місяці виконувалися умови:

- сумарний **MAPE** не перевищує 10 %;
- **MAPE** для кожної категорії не перевищує 20 %;
- середня абсолютна помилка (**MAE**) та зважена відносна помилка (**wMAPE**) фіксують реальну грошову й структуральну похибку.

Експериментальний план передбачає порівняння таких класів моделей:

- базові бенчмарки Naïve і Seasonal Naïve;
- статистичні ETS (Holt–Winters) та SARIMAX/SARIMA;
- адитивний Prophet з вбудованими святами;
- нелінійний XGBoost із лаг-ознаками;
- метод TSB для переривчастих категорій (Croston використано як додатковий еталон).

Навчання відбувається в схемі expanding-window CV із первинним вікном 24 місяці й кроком валідації один місяць, що імітує реальний режим експлуатації та зберігає причинно-часову послідовність.

Завдання дослідження полягає у визначенні, яка з перелічених моделей (чи їхня комбінація) найкраще задовольнить наведені KPI, а також у перевірці стабільності вибору на останніх шести місяцях тестового періоду.

Висновки до розділу 1

У першому розділі показано, що ручні або спрощені підходи до бюджетування не враховують сезонних і макроекономічних чинників, тоді як наявні фінтех-сервіси зосереджені переважно на візуалізації минулих даних.

Обґрунтовано доцільність використання методології CRISP-DM для системної розробки моделі бюджетного прогнозу; описано пул моделей, здатних працювати на коротких і частково переривчастих часових рядах; сформовано датасет із близько 4 000 транзакцій, приведений до єдиної еталонної структури й збагачений зовнішніми регресорами.

На підставі аналізу обрано чотири основні статистичні та машинні підходи й один спеціалізований метод для рідкісних витрат. Установлено кількісні критерії якості (MAPE, wMAPE, MAE) та експериментальну схему expanding-window. Надалі робота зосереджена на реалізації моделей, їх валідації за зазначеними KPI та інтеграції прогнозного модуля в BI-середовище Power BI, що створить практичне рішення для короткострокового планування бюджету за умов обмеженої історії даних і високої невизначеності зовнішнього середовища.

РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ МОДЕЛЕЙ ПРОГНОЗУВАННЯ

Особисті та малобізнесові бюджети формують короткі (≈ 24–48 місяців) часові ряди з нерідкими нульовими витратами та вираженою сезонністю. Класичні ARIMA/ETS-моделі потребують ≥ 2–3 сезонів для надійної оцінки параметрів, а за високої частки нулів (> 40 %) втрачають точність – у таких випадках застосовують Croston/TSB для моделювання «розміру» та «інтервалу» окремо. Водночас екзогенні чинники (курс USD/UAH, CPI, свята) систематично впливають на витрати, тому SARIMAX і Prophet дозволяють їх врахувати без втрати стабільності навіть на коротких серіях. Щоб об'єднати ці підходи, ми пропонуємо алгоритм rule-based відбору моделей за діагностикою кожної категорії та остаточну валідацію через expanding-window CV. У цьому розділі розглянемо математичні основи обраних методів, метрики оцінки та логіку автоматичного добору моделей.

2.1 Базові бенчмарки: Naïve та Seasonal Naïve

На першому кроці будь-якого експерименту прийнято порівнювати моделі з найпростішими «нульовими» підходами. У конкурсах M3 та M4 саме Naïve і Seasonal Naïve слугують стартовим орієнтиром, а понад 15 % поданих алгоритмів так і не перевершують їхню точність [39].

2.1.1 Метод Naïve

Для задач із дуже короткою історією без виразних трендів чи сезонності метод Naïve слугує найпростішим бенчмарком, який копіює останнє спостереження. Його прозорість й мінімальна кількість параметрів дозволяють швидко оцінити базову точність без складної підготовки даних. Ця стратегія **повністю детермінована** – не має жодних гіперпараметрів і не вимагає жодної оптимізації: прогноз просто копіює останнє значення ряду.

Найпростіша стратегія припускає, що майбутнє значення дорівнює останньому спостереженню (формула (1)).

$$\hat{y}_{t+h} = y_t,\ h = 1,2,\cdots \quad (1)$$

Наприклад, якщо витрата за березень становила 450 ₴, то прогноз на квітень становить ті ж 450 ₴. За потреби врахувати рівномірний тренд зростання чи спадання додають дрейф δ, як у формулі (2):

$$\hat{y}_{t+h} = y_t + h\delta \quad (2)$$

Це дає можливість врахувати постійне зростання чи спадання витрат без побудови складніших моделей.

Основна перевага Naïve-прогнозу – його абсолютна прозорість: щоб спрогнозувати значення, потрібна лише остання точка ряду, а довірчий інтервал легко отримати, додавши ±1,96 стандартні відхилення залишків [40]. Попри простоту, бенчмарк не такий слабкий: у конкурсі M4 на щоденних серіях він перевершив 18 % усіх поданих алгоритмів [39].

Недолік очевидний: метод повністю ігнорує тренд і сезонність, тому похибка з часом зростає до дисперсії самого ряду, а прогнозні інтервали «розповзаються» дедалі ширше.

З практичного погляду Naïve визначає мінімальну планку: модель уважають придатною лише тоді, коли її MAPE або sMAPE щонайменше на 10 % нижча за цю базову стратегію [41]. Такий поріг відсікає рішення, що дають лише косметичне покращення, і гарантує, що складніші алгоритми справді приносять відчутну користь.

Підсумок: Naïve – найпростіший бенчмарк без гіперпараметрів, приймаємо будь-яку складнішу модель лише при >10 % покращенні MAPE/sMAPE.

2.1.2 Метод Seasonal Naïve

Якщо дані демонструють чітку циклічність (річну або квартальну), Seasonal Naïve швидко відтворює сезонний профіль, просто копіюючи ці

значення з попереднього циклу. Цей підхід ідеальний для контролю «чистої» користі від будь-якої складності над збереженням регулярних коливань. Seasonal Naïve також **не містить гіперпараметрів** – порядок сезону (s) зазвичай фіксують за природним циклом даних (12 для річного, 4 для квартального), без жодної оптимізації.

Як правило, якщо дані містять стійку сезонність, то для рядів із чітким річним чи квартальним циклом логічно прогнозувати, що наступне значення повторює минулорічне у тій самій фазі сезону, для цього використовується формула (3):

$$\hat{y}_{t+h} = y_{t+h-s(\lfloor (h-1)/s \rfloor+1)}, \quad (3)$$

де s = довжина сезону.

Для Seasonal Naïve припускається, що кожний період точно повторює попередній. Наприклад, у щомісячному ряді з річним циклом (s=12) прогноз серпня 2025 року просто копіює фактичне значення серпня 2024 року. Цей простий прийом відтворює сезонний профіль і дає надзвичайно міцний бар'єр для коротких даних. Метод зберігає сезонний профіль без оцінки параметрів і часто слугує неперевершеним еталоном на дуже коротких вибірках [40]. По-друге, у конкурсі M4 на високочастотних серіях, що містили менше ніж 24 спостереження, Seasonal Naïve продемонстрував sMAPE на шість відсотків нижчу, ніж найкраща версія ETS, тобто перевершив складнішу модель на короткому горизонті [42].

Як і Naïve, Seasonal Naïve визначає мінімальну планку: статистична чи ML-модель має суттєво (≈ > 10 %) знижувати помилку щодо цього бенчмарку, щоби виправдати вищу складність [43].

Підсумок: Seasonal Naïve – базовий сезонний орієнтир без оптимізації, рекомендується перевершувати його помилку мінімум на 10 % перед застосуванням складніших алгоритмів.

2.2 Класичні статистичні моделі

2.2.1 ARIMA, SARIMA та SARIMAX

Класичні ARIMA-моделі підходять для часових рядів зі зрозумілою автокореляцією, дозволяючи поєднати прості лаги, диференціювання та ковзне середнє. Розширення SARIMA додає сезонні компоненти, а SARIMAX – екзогенні регресори (курс, свята тощо) без втрати стабільності.

Це сімейство авторегресійних моделей описує часовий ряд через власні запізнення, операції диференціювання та згладжений випадковий шум; у сезонній модифікації додається регулярний цикл, а у варіанті SARIMAX – ще й зовнішні регресори. Для стислого позначення використовують формулу (4)

$$SARIMA(p,d,q)\,(P,D,Q)_s \quad (4)$$

де p,d,q – порядки несезонної AR, I, MA-частин, P,D,Q – сезонної компоненти, s – довжина циклу.

Розгорнута операторна форма має формулу (5) (для одного коефіцієнта в кожному блоці)

$$(1-\varphi_1 B)(1-\Phi_1 B^s)(1-B)^d(1-B^s)^D y_t = (1+\theta_1 B)(1+\Theta_1 B^s)\varepsilon_t \quad (5)$$

де B – оператор лагу, φ,θ та Φ,Θ – поліноми AR/MA сезонного й несезонного рівнів; $\varepsilon_t \sim N(0,\sigma^2)$ – білий шум [44]. У SARIMAX до правої частини додається багаточлен $\beta(B)X_t$, що вводить вплив, наприклад, курсу USD/UAH чи інших факторів [44] [45].

Порядки $(p,d,q)\,(P,D,Q)$ зазвичай **автоматично обирають** за допомогою алгоритмів, які мінімізують інформаційний критерій (AICc/BIC), тому ручного перебору комбінацій як такого не потребується. Стаціонарність перевіряють диференціюванням доти, доки залишки не проходять тест Ljung–Box. Для фінансових рядів із ≤ 48 спостережень зазвичай вистачає $d \le 1$ та $D \le 1$; надмірне диференціювання зменшує інформацію про тренд. Діапазон параметрів часто скорочують автоматизованою функцією *auto_arima*, що перебирає комбінації та залишає модель із мінімальним AICc.

Наприклад, для «Комунальних» з річним циклом s=12 часто достатньо SARIMA$(1,1,1)(1,1,1)_{12}$, що дозволяє мінімізувати AICc та зберегти трендову структуру.

Коли даних щонайменше два сезони, підключення екзогенних регресорів у SARIMAX знижує MAPE на 5–15 % [45] порівняно з «чистою» SARIMA. Довірчі інтервали прогнозу обчислюють аналітично з дисперсії k-крокової похибки, що зручно для оцінки ризику недобору бюджету.

Для надійної оцінки сезонних параметрів потрібно щонайменше два повних цикли (≈ 24 місяці). за коротшої історії порядок моделі рекомендують спрощувати або фіксувати апріорно [21]. Модель доцільна, якщо кількість спостережень достатня, сезонність виразна, а періоди з нульовими витратами рідкі. За великої переривчастості перевагу надають легшим схемам.

Підсумок: SARIMA/SARIMAX доцільні при щонайменше двох повних сезонах і виразній сезонності; SARIMAX додає екзогенні регресори, якщо вони знижують MAPE щонайменше на 5 %.

2.2.2 ETS (Holt–Winters)

Метод експоненційного згладжування ETS придатний для даних із невеликою кількістю точок, оскільки оцінює лише рівень, тренд і сезонність без необхідності перевіряти стаціонарність. Завдяки цьому ETS дає стійкі прогнози вже за 24–30 спостережень і автоматично обирає адитивну або мультиплікативну форму.

Експоненційне згладжування трактує ряд як суму (або добуток) трьох латентних компонент – рівня ℓ_t, тренду b_t і сезонного індексу s_t – надаючи більшої ваги свіжим спостереженням. Конфігурацію коротко позначають формулою (6)

$$ETS(E,T,S)_s \quad (6)$$

де літери A чи M означають адитивну або мультиплікативну форму, N – відсутність компоненти, d – демпфований тренд.

Для класичної адитивної схеми $(A, A, A)_s$ використовується формула (7)

$$\begin{gathered} \ell_t = \alpha\,(y_t - s_{t-s}) + (1-\alpha)(\ell_{t-1} + b_{t-1}), \\ b_t = \beta\,(\ell_t - \ell_{t-1}) + (1-\beta)\,b_{t-1}, \\ s_t = \gamma\,(y_t - \ell_t) + (1-\gamma)s_{t-s}, \\ \hat{y}_{t+h} = \ell_t + h\,b_t + s_{\,t+h-s[(h-1)\bmod s+1]}, \end{gathered} \tag{7}$$

де $\alpha, \beta, \gamma \in (0,1]$ – коефіцієнти згладжування [46]. Якщо тренд поступово сповільнюється, замість звичайного b_t використовують демпфовану версію $b_t^* = \phi b_t$ з $\phi \in (0,1)$, що запобігає завеликим довгостроковим прогнозам.

Коефіцієнти α,β,γ (і, за потреби, демпфінговий параметр ϕ) **оцінюють максимізацією правдоподібності** в рамках стан-простору, без додаткової ручної тонкої настройки. Параметри оцінюють максимумом правдоподібності; конфігурацію ETS автоматично вибирає функція *autoes* за мінімальним AICc. Модель стійка вже на 24–30 точках і не вимагає стаціонарності, проте чутлива до довгих послідовностей нулів: рівень ℓ_t стрімко спадає, а прогноз знижується до нуля; тоді варто перейти до переривчастих методів Croston/TSB [46] [47]. Коли сезонний сигнал слабкий ($R^2_{season} < 0.3$, доцільно застосувати $ETS(A, A, N)$ або навіть Theta-метод – статистичний еталон, що переміг у конкурсі M3 і добре працює на дуже коротких рядах.

Моделі ETS забезпечують довірчі інтервали через стан-простір і Калманів фільтр, а Box–Cox-перетворення λ-ступеня часто застосовують до вихідних даних для стабілізації дисперсії перед оцінюванням; у бібліотеках *forecast* (R) і *statsforecast* (Python) ці опції вмикаються автоматично.

Підсумок: ETS – надійне згладжування для 24–30 точок без перевірки стаціонарності, але чутливе до довгих нульових блоків (>40 %), після чого слід переходити до переривчастих методів.

2.3 Методи для переривчастих рядів

У категоріях із тривалими нульовими відрізками (наприклад, страхування чи великі покупки) ARIMA та ETS часто дають прогнози, близькі до нуля, оскільки їхній рівень постійно оновлюється нулями. Коли частка нульових > 40

%, класична статистика провалюється, тож застосовують переривчасті методи. Щоби розділити розмір витрачених коштів і інтервал між транзакціями, використовують алгоритми Croston і TSB. Обидва ведуть дві змінні: середній чек та інформацію про ймовірність появи нової події [48] [49].

2.3.1 Метод Croston

Для переривчастих рядів із періодами без витрат понад 40 % підходить метод Croston, що окремо моделює середній чек і середній інтервал між транзакціями. Така двокомпонентна структура дозволяє отримати базовий прогноз, навіть якщо події трапляються нерегулярно.

Croston – це простий підхід до прогнозування рідкісних витрат, який незалежно оцінює середній розмір транзакції та середній інтервал між ними, а результатом є їхнє відношення. Він працює найкраще, коли операції трапляються хоч інколи, але нерегулярно [50].

Для скороченого представлення використовується формула (8)

$$Croston(\alpha),\ \alpha \in (0,1] \quad (8)$$

де α – це коефіцієнт згладжування, який визначає, наскільки сильно нове спостереження впливає на поточну оцінку.

Повне представлення математично описується формулою (9)

$$\begin{gathered} z_t = \alpha\, y_t + (1-\alpha)\, z_{t-1}, \\ p_t = \alpha\, \tau_t + (1-\alpha)\, p_{t-1}, \quad (9) \\ \hat{y}_{t+1} = z_t / p_t \end{gathered}$$

Якщо в місяці трапилась покупка, модель оновлює середній чек z_t і середній інтервал τ_t, то наступного місяця логіка моделі буде такою: «в середньому чек такий, а покупка трапляється раз на τ_t місяців», тобто прогноз = чек / інтервал.

У Croston однаковим α згладжуються і середній розмір транзакції z_t, і середній інтервал τ_t. Якщо α ближче до 1, модель швидше реагує на зміни (нові

y_t одразу сильно коригують z_t, і τ_t); якщо ближче до 0, історичні значення мають більшу вагу і згладжування довше «тягне» старі очікування.

Початкові значення z_0 і p_0 зазвичай обирають рівними першому ненульовому обсягу транзакції та відстані (у місяцях) від початку спостережень до цієї події, що дозволяє одразу запустити згладжувальні формули. Коефіцієнт згладжування α підбирають автоматично за мінімізації AICc або ж фіксують у межах приблизно 0,1–0,3, щоб балансувати швидкість адаптації й стійкість оцінок. Щоби усунути властиве класичному Croston-методу позитивне зсування прогнозу, застосовують SBA-поправку: базову оцінку z_t/p_t множать на фактор $(1 - \alpha/2)$, що знижує надмірне завищення оцінок [48]. Дослідження показали, що в реальних переривчастих рядах додаток SBA-корекції знижує середню упередженість прогнозу майже на 20 %. Для багатокрокових прогнозів (оцінюють значення не лише для наступного періоду, а відразу на кілька кроків уперед) зазвичай використовують лінійне масштабування: однокроковий прогноз множать на число кроків h, що спрощує побудову неточних, проте швидких оцінок витрат на горизонті більше одного місяця.

Недоліком методу є те, що він повільно реагує на зникнення витрат. Якщо в один момент витрати певної категорії припиняються, він може і далі продовжувати прогнозувати, що витрати є.

Підсумок: Croston – простий двокомпонентний метод для рідкісних витрат, підходить при переривчастості >40 %, зате сповільнено реагує на повне зникнення подій. Водночас Syntetos & Boylan (2005) показали, що застосування SBA-поправки знижує середню упередженість прогнозу близько на 20 % порівняно з оригінальною схемою Croston [51].

2.3.2 Метод TSB (Teunter–Syntetos–Babai).

TSB розвиває ідею Croston, додаючи окреме згладжування ймовірності події та оновлення розміру чека лише при фактичній операції [52]. Це дозволяє прогнозу миттєво реагувати на раптові обнулення й зміну частоти транзакцій.

Для стислого позначення використовують формулу (10)

$$TSB(\alpha, \beta),\ \alpha, \beta \in (0,1] \quad (10)$$

де α та **β** – це коефіцієнти згладжування.

Повне представлення математично описується формулою (11)

$$p_t = \alpha\, I_{\{y_t>0\}} + (1-\alpha)\, p_{t-1},$$
$$z_t = \begin{cases} \beta y_t + (1-\beta) z_{t-1}, & y_t > 0, \\ z_{t-1}, & y_t = 0, \end{cases} \quad (11)$$
$$\hat{y}_{t+1} = p_t z_t$$

α задає швидкість оновлення **ймовірності** появи ненульової події p_t. Високе α означає, що один місяць без витрат сильно зменшить оцінку p_t, а місяць з витратами – швидко підніме її.

β керує згладжуванням **розміру** транзакції z_t. Якщо β велике, нове ненульове y_t лягає в основу майже одразу; якщо маленьке, середній чек змінюється поступово.

У TSB-моделі кожного місяця насамперед перевіряється, чи відбулася транзакція: у разі ненульового витрачання оновлюється середній чек, а якщо витрат не було, знижується ймовірність події. Завдяки цьому прогноз миттєво реагує на обнулення попиту, не чекаючи кумулятивної помилки.

Практична реалізація починається з ініціалізації: ймовірність p_0 встановлюють як частку місяців із витратами у історії, а початковий чек z_0 – рівним першому ненульовому значенню. Коефіцієнти згладжування α і β зазвичай автоматично добирають через мінімізацію AICc або крос-валідацію, що дозволяє знайти баланс між чутливістю моделі та стабільністю оцінок. Для оцінки точності прогнозів підходять метрики MASE або sMAPE, оскільки стандартний MAPE некоректно працює в присутності нульових спостережень. У популярних пакетах це реалізовано «з коробки», наприклад у Python можна використати statsforecast.TSB, яка автоматично виконує ініціалізацію та пошук оптимальних параметрів.

У експериментах Teunter et al. (2011) показали, що TSB дає на 20–30 % нижчий MASE проти класичного Croston у сценаріях з раптовими обнуленнями.

Підсумок: TSB – удосконалена Croston-схема з окремим згладжуванням імовірності події, дає на 20–30 % нижчий MASE у сценаріях із раптовими обнуленнями. TSB – ефективний інструмент для категорій витрат, де частота подій змінюється у часі або може тимчасово припинятися, адже він забезпечує швидке коригування ймовірності транзакції та стабільніші прогнози в умовах високої переривчастості.

2.4 Сучасні ML-підходи

2.4.1 Prophet (Taylor & Letham, 2018)

Prophet – адитивна байєсівська модель, що поєднує piece-wise тренд, гармонічну сезонність та довільні ефекти свят. Вона підходить для швидкого розгортання на бізнес-даних з нерегулярними викидами та обмеженою історією. Усі параметри моделі (тренд-злами, гармонічні коефіцієнти, вплив свят) **оцінюються** в єдиному байєсівському фреймворку через стохастичний градієнтний спуск, тому додаткового гіперпараметричного перебору не потребується.

Формально прогноз записується за допомогою формули (12)

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t, \quad (12)$$

де g(t) задає тренд (лінійний або з насиченням), $s(t) = \sum_{k=1}^{K}[a_k \sin(2\pi kt/s) + b_k \cos(2\pi kt/s)]$ описує сезонність із періодом s, h(t) відображає ефекти свят і подій, а $\varepsilon_t \sim N(0, \sigma^2)$ – шум. Параметри оцінюють у байєсівській рамці через стохастичний градієнтний спуск, автоматично визначаючи кількість точок «зламу» тренду та активацію святкових ефектів [53].

Prophet особливо корисний на коротких серіях (24–36 спостережень), оскільки стійкий до пропусків і викидів, легко вбудовує довільні екзогенні свята й у багатьох прикладах перевершує ETS/ARIMA за sMAPE на 10–12 % [54]. Проте модель може «перетягнути» сезонність, якщо число гармонік K вибране

занадто великим, та втрачає перевагу на категоріях без явної циклічності, де її прогноз зводиться до ковзного середнього.

Підсумок: Prophet – байєсівська адитивна модель зі зламами тренду та святковими ефектами, стійка до пропусків і викидів, перевершує ETS/ARIMA на 10–12 % за sMAPE на коротких серіях.

2.4.2 XGBoost-lags (градієнтний бустинг)

XGBoost-lags перетворює прогнозування часових рядів на задачу регресії, використовуючи лаги та екзогенні ознаки як фічі. Такий підхід ефективний для виявлення нелінійних взаємодій і дозволяє «глобально» навчати модель на багатьох серіях одночасно. XGBoost-lags **має низку гіперпараметрів** (кількість дерев, learning rate η, глибина дерев, регуляризацію тощо), які зазвичай оптимізують через сітковий або випадковий пошук (Grid/RandomSearch) або AutoML-інструменти.

Для математичного представлення використовується формула (13)

$$\hat{y}_{t+1} = f_\Theta(x_t),\ x_t = [y_t, y_{t-1}, \dots, y_{t-L}, z_t, \dots], \quad (13)$$

де f_Θ – ансамбль з M дерев рішень, $f_\Theta(x) = \sum_{m=1}^{M} \eta T_m(x; \theta_m)$, η – learning rate, L – глибина лагів, а z_t включає екзогенні ознаки.

На практиці L (макс. глибина лагів) зазвичай беруть ≤ 12 для місячних даних – інакше модель починає “вловлювати” випадковий шум.

У конкурсі M5 Accuracy найкращі рішення на базі XGBoost-lags знизили середню wRMSE на понад 14 % від найкращого статистичного бенчмарку [33].

Цей підхід дозволяє моделювати складні нелінійні взаємодії та швидко додавати календарні, цінові чи промо-фічі без спеціальної трансформації. Навчання «глобальної» моделі на всіх категоріях одразу підвищує стабільність прогнозів, коли в кожної серії менше ніж 30 точок. Водночас XGBoost-lags потребує ретельного вибору лагів і гіперпараметрів, а довірчі інтервали будується лише бутстрепом чи quantile-loss-функціями, що може значно збільшити обчислювальні витрати.

Підсумок: XGBoost-lags – нелінійний градієнтний бустинг із лаг-фічами й екзо-ознаками, знижує wRMSE на ≥ 14 %, але потребує оптимізації гіперпараметрів і бутстреп-довірчих інтервалів. Метод XGBoost-lags доцільний, коли разом із короткою історією доступні обґрунтовані екзогенні ознаки й потрібне моделювання нелінійностей, особливо у випадках слабкої чи складної сезонності.

2.5 Метрики та валідація

У будь-якій системі прогнозування остаточне рішення про застосування тієї чи іншої моделі ухвалюють на підставі **об'єктивної оцінки її якості**. Саме тут на сцену виходять **метрики** – кількісні показники, які вимірюють, наскільки близько прогнози лежать до реальних значень, і допомагають зрозуміти, як модель поводиться в різних ситуаціях: коли дані містять нулі, коли є різкий сплеск витрат або коли потрібно пріоритезувати одні категорії над іншими. Проте жодна метрика сама по собі не дає повної картини, тому важливо підібрати **комплементарний набір** та перевірити їхню поведінку у процесі моделювання. А для уникнення «витоку» інформації з майбутнього імітують реальне оновлення модельного зразка за допомогою **expanding-window cross-validation**, яке гарантує, що оцінка метрик відбувається лише на даних, недоступних моделі під час навчання.

2.5.1 Ключові метрики для бюджетних часових рядів

У контексті прогнозування витрат із короткою історією (≈ 24–48 міс.), нерідкими нульовими значеннями та вираженою сезонністю переважно використовують чотири взаємодоповнювальні метрики.

MAPE (Mean Absolute Percentage Error).

MAPE вимірює середню відносну похибку прогнозу у відсотках від фактичного значення, відповідаючи на питання: «наскільки відсотково мій прогноз в середньому відхиляється від реальних витрат?».

Формально виражається формулою (14):

$$\text{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100\%,\ y_t > 0, \quad (14)$$

де y_t – фактичне, $\widehat{y_t}$ – прогнозоване значення.

Ця метрика інтуїтивно зрозуміла бізнес-користувачам і часто застосовується як перший індикатор точності [55]. Водночас при $y_t \approx 0$ метрика нестійка і «вибухає», а через асиметрію недооцінка й переоцінка дають різні відсотки помилки.

sMAPE (Symmetric MAPE).

Щоби позбутися нестабільності MAPE при y_t близьких до нуля, sMAPE нормує похибку на середнє між фактичним та прогнозом (15):

$$\text{sMAPE} = \frac{100\%}{T}\sum_{t=1}^{T}\frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2} \qquad (15)$$

Завдяки такому підходу навіть при нульових y_t знаменник не обнуляється, а значення sMAPE лишається в межах $0\% - 200\%$ [56].

MASE (Mean Absolute Scaled Error)

Переривчасті ряди з численними нулями потребують порівняння з найпростішим Naïve-прогнозом. MASE шкалює абсолютну помилку на середню величину однокрокової зміни ряду (16):

$$\text{MASE} = \frac{\frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t|}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|} \qquad (16)$$

MASE має однакову шкалу для серій різного масштабу, нечутлива до масштабу та стабільна за наявності нульових спостережень, тому особливо корисна для переривчастих витрат [57].

wRMSE (Weighted Root Mean Squared Error)

У випадках комбінування категорій різної ваги іноді доцільно приділяти більше уваги критичним чи дорожчим статтям витрат. Для цього

використовують метрику wRMSE, що дає більшу вагу категоріям із більшими сумами витрат або стратегічною важливістю.

Метрика має таку формулу (17):

$$\text{wRMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} w_t \ (y_t - \hat{y}_t)^2}, \qquad w_t \geq 0, \ (17)$$

де w_t – довільні ваги (наприклад, пропорційні обсягу витрат).

У конкурсі M5 саме ця зважена версія RMSE визнана стандартом для великих чисел часових рядів [58].

У більшості практичних задач бюджетного прогнозування саме цей набір метрик забезпечує баланс між інтуїтивністю (MAPE/sMAPE), стійкістю до нульових або малих значень (sMAPE, MASE) та можливістю пріоритезувати важливі категорії (wRMSE).

2.5.2 Додаткові метрики

Коли бізнес-вимоги виходять за межі точкового прогнозу й включають оцінку типових помилок, великих стрибків чи невизначеності, то базового набору недостатньо. У такому випадку стають у пригоді метрики MAE, RMSE, CRPS і Coverage.

MAE (Mean Absolute Error)

Є суто абсолютною мірою, MAE легша у трактуванні і не збільшує вагу далеких викидів. Корисна, коли потрібно зменшити вплив великих викидів і отримати уявлення про «типову» помилку. Формула (18):

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t| \qquad (18)$$

RMSE (Root Mean Squared Error)

Ця метрика виділяє великі помилки завдяки квадратичному зважуванню. Вона надає квадратичну вагу великим відхиленням, що допомагає виділити ситуації з раптовими стрибками витрат. Обчислюється формулою (19):

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \quad (19)$$

CRPS (Continuous Ranked Probability Score)

CRPS оцінює якість усього прогнозного розподілу, коли модель повертає прогнозний розподіл, а не одиничне число [59].

Coverage / Interval Score

Ці метрики використовуються для перевірки коректності довірчих інтервалів (наприклад, 80 % чи 95 %). Вони дають змогу контролювати, чи справді інтервали містять фактичні значення з потрібною часткою.

У підсумку, для більшості бюджетних часових рядів досить чотирьох ключових метрик – MAPE та sMAPE для оцінки відносної похибки, MASE для коректного порівняння з простим Naïve-прогнозом і wRMSE для виділення стратегічно важливих категорій. Вони забезпечують баланс між інтуїтивністю, стійкістю до нульових або малих значень і можливістю врахувати пріоритети бізнесу. Однак у ситуаціях, коли необхідно зосередитися на типових (MAE) чи навпаки дуже великих (RMSE) відхиленнях, а також при роботі з ймовірнісними прогнозами або довірчими інтервалами (CRPS, Coverage), доцільно додати відповідні додаткові метрики, щоб отримати повну картину точності та надійності моделі.

2.5.3 Валідація прогнозів: expanding-window cross-validation

Для відтворення процесу постійного оновлення та перевірки, чи модель не «заглядає в майбутнє», застосовують **expanding-window cross-validation –** поступове розширення навчальної вибірки. Цей метод максимально наближує процедуру тестування до реального циклу експлуатації [60].

Валідація складається з трьох кроків:

1. **Стартове вікно.** Фіксуємо перші m місяців (наприклад, 24 місяці) як початкову тренувальну вибірку.

2. **Крокуюче розширення.** Кожен новий місяць додається до тренувальної вибірки, після чого модель перенавчається і робить прогноз на h-місячний горизонт.

3. **Агрегація помилок.** На кожному кроці обчислюємо MAPE, sMAPE, MASE і wRMSE, а потім узагальнюємо результати по всіх ітераціях.

Такий підхід не лише **перевіряє стабільність** моделі на коротких серіях, але й **віддзеркалює реальний процес** постійного оновлення прогнозів за реального використання.

2.6 Алгоритм автоматичного вибору моделі для кожної категорії

Автоматизоване прогнозування сотень-тисяч бюджетних статей неможливо підтримувати вручну: для кожної категорії потрібно обрати модель, яка найкраще відповідає її статистичним рисам, і регулярно перевіряти, чи це рішення не застаріло. Поведінка різних категорій істотно відрізняється, а отже й набір адекватних моделей змінюється від серії до серії та з часом.

З цієї причини пропонується алгоритм на основі правил, що спочатку проводить діагностику статистичних властивостей кожної часової серії, а потім, на основі чіткого набору порогових правил, обирає попередню модель. Остаточна валідація відбувається через розширювальну крос-валідацію (expanding-window CV), яка гарантує, що рішення не ґрунтується на інформації з майбутнього.

2.6.1 Діагностичні показники

Для кожної категорії ми обчислюємо чотири ключові характеристики. Вони є простими, але достатньо інформативними.

Довжина історії T.

Це загальна кількість місячних спостережень (20).

$$T = |\{y_t\}_{t=1}^{T}| \qquad (20)$$

Коли $T \geq 24$, статистичні сезонні моделі вже можуть надійно оцінити свої параметри [61]. При коротшій історії доцільно починати з простіших підходів, наприклад Naïve моделям.

Переривчастість (переривчастість) Z.

Для того, щоби визначити переривчастість часового ряду, потрібно кількість нульових спостережень поділити на довжину ряду (21). Таким чином знайдемо частку нульових спостережень.

$$Z = \frac{\sum_{t=1}^{T} \mathbb{I}(y_t=0)}{T} \quad (21)$$

Якщо $Z > 0.4$, тобто понад 40 % місяців без витрат коштів, класичні ARIMA/ETS-підходи можуть занижувати рівень і ставати некоректними. Для таких випадків доцільніше застосовувати методи Croston або TSB [62].

Сила сезонності S.

S – коефіцієнт детермінації R^2 від регресії на індикаторних змінних місяців (22) (23).

$$y_t = \beta_0 + \sum_{m=1}^{11} \beta_m D_{m,t} + \varepsilon_t, \quad (22)$$

$$S = R^2 \geq 0.3 \quad (23)$$

де $D_{m,t}$ – місячні індикаторні змінні.

Тут $D_{m,t} = 1$, якщо спостереження t припадає на місяць m, інакше 0. Сезонність вважаємо вираженою за $S = R^2 \geq 0.3$.

Такий поріг рекомендований у практичних посібниках зі статистики сезонних рядів [63].

Наявність екзогенних чинників.

Перевіряємо, чи зовнішні регресори $X_t^{(k)}$ (курс USD/UAH, індекс споживчих цін, бінарні «свята» тощо) статистично значущо впливають на серію. Для цього потенційні регресори включають у допоміжну SARIMAX-регресію (24).

$$y_t = \alpha + \sum_{k=1}^{K} \gamma_k X_t^{(k)} + \varepsilon_t \quad (24)$$

Якщо хоча б один коефіцієнт γ_k статистично значущий ($p < 0.05$) і AICc моделі з екзогенами зменшується порівняно з моделлю без них, фіксуємо наявність релевантних зовнішніх пояснювачів [64].

Зібравши **T**, **Z**, **S** та інформацію про екзогени, отримаємо повний статистичний «профіль» серії, достатній для подальшого автоматизованого вибору моделі.

2.6.2 Правила попереднього відбору

Після діагностики серія послідовно перевіряється на відповідність п'ятьом взаємовиключним сценаріям. Порогові значення наразі беруться умовні, у розділі 3 вони будуть перевірені та оновлені на основі кластеризації.

1. Переривчасті ряди ($\boldsymbol{Z > 0.4}$). Навчаємо та оцінюємо моделі Croston та TSB. Якщо середня похибка TSB при розширювальній CV не перевищує 95 % похибки Croston ($\boldsymbol{MASE_{TSB} \leq 0.95} \cdot \mathrm{MASE}_{\mathrm{Croston}}$,), обираємо TSB; інакше лишається Croston.

2. Коротка, але сезонна історія ($\boldsymbol{T < 24}$ і $\boldsymbol{S \geq 0.3}$). У ролі стартового орієнтира використовується Seasonal Naïve. Якщо ETS демонструє зниження sMAPE щонайменше на 10 % порівняно з сезонною наївною моделлю, перевага віддається ETS.

3. Короткі й несезонні серії з екзогенами ($\boldsymbol{T < 36}$, $\boldsymbol{S < 0.3}$, є релевантні екзогени). Першим кандидатом є Prophet – його затверджують, якщо він забезпечує зменшення sMAPE щонайменше на 10 % від ETS.

4. Достатньо довгі сезонні серії ($\boldsymbol{T \geq 24}$, $\boldsymbol{S \geq 0.3}$, $\boldsymbol{Z \leq 0.4}$). Застосовуємо автоматичний підбір порядків SARIMAX. SARIMAX утверджується, якщо його MAPE менша на 5 % MAPE ETS.

5. Багато екзогенних ознак і можливі нелінійності

Якщо регресори статистично значущі та очікуються нелінійні взаємодії, конкурує XGBoost-lags. Перевіряємо XGBoost-lags у порівнянні з найкращим

статистичним суперником – якщо його wRMSE зменшується щонайменше на 10 %, обираємо XGBoost.

2.6.3 Підтвердження через expanding-window CV

Попередній вибір вважається дійсним, лише якщо модель проходить розширювальну CV: початково береться вікно з перших m місяців, потім по черзі додається по одному спостереженню, кожного разу перенавчається модель і прогнозується горизонт h. За всіма ітераціями акумулюється значення метрик (MAPE, sMAPE, MASE, wRMSE) і обчислюються їх середні. Якщо жодна з метрик не демонструє покращення понад порогові значення (10 % для відносних, 5 % для абсолютних), алгоритм повертається до наївного прогнозу (Seasonal Naïve або Naïve) та позначає серію для ручного перегляду.

2.6.4 Автоматичне оновлення

Щокварталу для кожної категорії повторно обчислюється T, Z, S та тестується значущість екзогенних змінних. У разі зміни поведінки ряду – скажімо, сезонність зменшилася нижче 0.3 або частка нулів зросла понад 0.4 – алгоритм автоматично переобирає та переоцінює модель, зберігаючи її актуальність без участі аналітика.

Завдяки комбінації прозорих статистичних індикаторів і чітко сформульованих порогових правил запропонований підхід дає можливість швидко та відтворювано призначати найдоцільнішу модель для кожної бюджетної категорії, а регулярна розширювальна валідація запобігає деградації якості при зміні статистичних властивостей даних.

Висновки до розділу 2

У *розділі 2* було детально розглянуто методи прогнозування бюджетних часових рядів із обмеженою історією, чітко вираженою сезонністю та значною часткою нульових спостережень. Зокрема, описано дві прості базові стратегії – Naïve і Seasonal Naïve – які забезпечують мінімальний рівень точності та слугують відправною точкою для подальших покращень. Далі представлено класичні статистичні моделі ARIMA/SARIMA (а також SARIMAX із екзогенними регресорами) і ETS, які вимагають наявності щонайменше двох повних сезонних циклів і демонструють потужність у моделюванні тренду, автокореляції та сезонності за умови низької переривчастості.

У разі коли понад 40 % спостережень є нульовими, класичні алгоритми систематично занижують прогноз, тому обґрунтовано застосування переривчастих методів Croston і TSB. Ці підходи розділяють моделювання розміру транзакцій і інтервалів між ними, що суттєво знижує упередженість і підвищує якість прогнозів. Крім того, розглянуто сучасні машинно-навчальні інструменти: Prophet із байєсівським трендом, гармоніками сезонності та врахуванням святкових ефектів, а також XGBoost із лаговими змінними й екзогенними ознаками для виявлення складних нелінійних взаємозв’язків.

Для об’єктивного порівняння моделей обрано чотири взаємодоповнювальні метрики (MAPE, sMAPE, MASE, wRMSE) та процедуру expanding-window cross-validation, що гарантовано оцінює прогнози лише на «незнайомих» даних і відображає реальний процес поступового оновлення інформації. Синтез усіх компонентів – від теоретичного обґрунтування до rule-based диспетчера моделей та циклічної перевірки на нових спостереженнях – формує відтворювану, прозору й адаптивну систему прогнозування. Така архітектура здатна автоматично вибирати найрелевантніший підхід для кожної бюджетної категорії й підтримувати його актуальність без постійного ручного втручання.

РОЗДІЛ 3. ПІДГОТОВКА ДАНИХ І РЕАЛІЗАЦІЯ АЛГОРИТМІВ ДЛЯ ПРОГНОЗУВАННЯ КОРОТКОСТРОКОВИХ ВИТРАТ

3.1 Вибір інструментів реалізації

Підготовка даних та експериментальний конвеєр виконувались у Python 3.10 у середовищі Jupyter Notebook. Така зв'язка надає три переваги, критичні саме для задач короткострокового бюджетного прогнозування.

Неперервна документація процесу. Код, текстові пояснення та проміжні діаграми зберігаються в єдиному файлі .ipynb, що спрощує рев'ю й відтворення результатів.

Гнучке керування ресурсами. Дослідницькі обчислення (підбір параметрів SARIMAX, навчання XGBoost) можна виконувати по-клітинково, оперативно коригуючи обсяг даних або гіперпараметри.

Широка екосистема бібліотек. Бібліотеки **pandas, numpy, datetime** забезпечують базове оброблення та агрегацію транзакцій; **statsmodels, pmdarima** надають реалізації класичних статистичних моделей разом із діагностикою залишків; бібліотеки **prophet, xgboost** надають реалізації алгоритмів сучасного ML; бібліотеки **holidays, workalendar** дозволяють автоматично формувати календарні регресори; бібліотеки **matplotlib, seaborn, plotly** забезпечують контрольні візуалізації якості очищення й агрегованих рядів.

Усі необхідні бібліотеки зафіксовані у requirements.txt.

Для роботи з даними створено з Jupyter ноутбуки:

01 data_preparation.ipynb – містить операції з очистки та підготовки даних, які будуть використовуватися у моделюванні. Зберігає підготовлені дані у файлах transactions.csv та transactions_with_exog.csv;

02 data_analysis.ipynb – файл для аналізу транзакцій, які завантажуються з попередньо підготовлених файлів;

03 data_modeling.ipynb – файл для створення та аналізу моделей.

3.2 Підготовка і обробка даних

3.2.1 Огляд конвеєру обробки даних

Робочий конвеєр складається з чотирьох логічних фаз.

Спершу сирі виписки Monobank і PrivatBank завантажуються з джерел та обробляються, щоби вони були придатні для подальшої обробки.

Після цього кожне з джерел попередньо обробляється та зводиться до єдиної структури даних: прибираються зайві стовпці, перейменовуються, створюється окремий стовпець для категорії транзакції (для monobank – на основі коду MCC [38], а для транзакцій Приватбанку – категорія доступна, якщо вказана у описі транзакції). Також вказується тип транзакції (дохід чи витрата), а також банк.

Коли дані зведені до однакової структури даних, води об’єднуються у єдину таблицю (таблиця 3.1). Після цього, за допомогою допоміжних таблиць, створених вручну на основі опису транзакцій кожна транзакція категоризується – спершу вже наявні категорії розподіляються між більшими групами категорій на основі Excel таблички *custom categories.xlsx* (рис 3.1), а після цього додаткова корекція категоризації виконується з використанням Excel таблички *description_category.xlsx* (рис 3.2). Також відфільтровуються транзакції, які є внутрішніми переказами між власними картками. Після цих операцій підготовлені транзакції зберігаються у файлі *transactions.csv*.

На основі створеного файлу *transactions.csv* здійснюється доповнення датасету екзогенними ознаками. Додаються екзогенні змінні, такі як курс валют, свята, індекс цін. Дані з додатковими змінними зберігаються у файл *transactions_with_exog.csv*. Це є останнім етапом. Подальша обробка даних, така як агрегація за місяцями та категоріями, відбувається вже безпосередньо перед моделюванням даних.

Таблиця 3.1 – **Структура даних після об’єднання транзакцій двох банків**

№	Назва поля	Опис
1	datetime	Дата та час проведення транзакції у форматі ДД.ММ.РРРР ГГ:ХХ; відображає момент списання або зарахування коштів.
2	description	Опис транзакції – назва торговельної точки чи сервісу, адреса, тип операції (покупка, повернення, переказ тощо).
3	amount_in_uah	Сума транзакції у гривнях (валюта карток)
4	amount_in_currency	Сума транзакції у валюті операції – еквівалент суми операції в гривні, перерахований банком за внутрішнім курсом.
5	currency	Код валюти операції
6	bank	Назва банку, з якого отримано дані про транзакцію
7	type	Вид транзакції (income/spend)
8	common_category	Стандартизована назва категорії, до якої відноситься транзакція

category	*common category (custom)*
Переказ на свою картку	Внутрішні перекази
Будматеріали	Дім та ремонт
Дім та ремонт	Дім та ремонт
Для дому	Дім та ремонт
Меблі	Дім та ремонт
Ремонт/Будівництво	Дім та ремонт
Товари для дому	Дім та ремонт
Косметика	Догляд: Гігієна
Краса	Догляд: Гігієна
Акваріуми. Дельфінарії	Дозвілля
Канцтовари	Дозвілля
Кіно	Дозвілля
Кінотеатри	Дозвілля
Книгарні	Дозвілля
Книги та канцтовари	Дозвілля
Прокат відео	Дозвілля
Розваги	Дозвілля
Розваги та спорт	Дозвілля
Спорт	Дозвілля
Спортклуби	Дозвілля
Товари для спорту	Дозвілля
Автоплатіж	Донати
Благодійність	Донати
Благочинність	Донати
Фонди та організації	Донати
Кур'єрська служба	Доставка
Кур'єрські та поштові послуги	Доставка
Транспортування. Доставка	Доставка
Заощадження	Заощадження
Аптека	Здоров'я
Аптеки	Здоров'я
Здоров'я й краса	Здоров'я

Рис 3.1 – Структура таблиці, що використовується для зведення різних назв категорій витрат до єдиного уніфікованого переліку

category	description contains
Покупки	Оплата в інтернет-магазині
Покупки	Оплата послуг через Приват24
Покупки	Оплата товарів / послуг через інтернет
Покупки	Переказ на картку ПриватБанку через додаток Приват24
Внутрішні перекази	Катерина Немченко
Внутрішні перекази	На свою картку
Внутрішні перекази	ПОКУПКА, Маркетплейси
Внутрішні перекази	Переказ на свою карту
Внутрішні перекази	Максим А.
Внутрішній переказ	На свою картку
Догляд: Гігієна	Kosa-eco
Догляд: Гігієна	maysternya myla
Догляд: Гігієна	МАЙСТЕРНЯ МИЛА
Догляд: Гігієна	Мила
Догляд: Гігієна	Федоренко Ірина Іванівна
Догляд: Гігієна	Фінкільштейн Ольга Валеріївна
Догляд: Гігієна	Чистенько
Догляд: Послуги	ESTET CENTR LOTOS
Догляд: Послуги	Білан
Догляд: Послуги	GOMZYAK STUDIO
Догляд: Послуги	MarafetStudio
Догляд: Послуги	Лотос
Догляд: Послуги	Брови
Догляд: Послуги	манікюр
Догляд: Послуги	стрижка
Дозвілля	Biletik
Дозвілля	Concert.ua
Дозвілля	Teatr
Дозвілля	Ботанчний сад
Дозвілля	Музей скла
Дозвілля	Парк «Муромець»
Дозвілля	Протасів Яр

Рис 3.2 – Структура таблиці, що використовується більш точної класифікації транзакцій за категоріями на основі їх описів

3.2.2 Підготовка сирих даних Monobank і PrivatBank

На першому етапі для обох джерел було реалізовано завантаження та приведення до єдиного формату. Для Monobank використовувався CSV-експорт. Для PrivatBank використовувася файл PDF, що конвертувався у CSV. Кожен файл читається з парсингом datetime, після чого:

- відкидаються технічні стовпці (наприклад, внутрішні ідентифікатори транзакції);

- змінюються назви полів на єдиний набір (datetime, amount_in_uah, amount_in_currency, currency, description, category, bank);
- створюється булевий прапорець type, що розрізняє витрати й доходи за знаком суми.

На цьому етапі дані Monobank доповнені категоріями на основі датасету з переліком категорій MCC [38]. У даних PrivatBank категорії були отримані з описів транзакцій, однак категорії доступні не для всіх рядків.

Таким чином отримано дві “чисті” таблиці, готові до подальшої консолідації.

3.2.3 Об’єднання даних у єдиний датафрейм

Після підготовки окремих витягів обидва набори даних зливаються в один DataFrame на основі спільних полів.

Перед злиттям:

- вирівнюються типи даних (усі дати в datetime64[ns], суми в float);
- здійснюється дедуплікація – рядки з тотожними значеннями datetime, amount_uah, description видаляються.

У результаті формується єдиний набір, в якому кожен рядок містить дату, банк, суму в UAH, оригінальну валюту, текстовий опис і базову інформацію про тип операції (таблиця 3.1).

3.2.4 Категоризація та фільтрація транзакцій

Для забезпечення коректності моделювання транзакції розбиваються на три групи: власне витрати, внутрішні перекази й повернення.

Надалі категоризація здійснюється з використанням створених вручну файлів-словників, що уніфікують назви категорій між банками, укрупняють категорії, а також коригують категорії на основі ключових слів з описів

транзакцій, якщо первинна категоризація не була коректною. Для цього використовуються файли *custom category.xlsx* та *description_category.xlsx*

Після цього рівень невизначених категорій знижується до <10 %, а точність на вибірці 500 транзакцій перевищує 92 %.

Також на цьому етапі видалені транзакції, по яких є пізніше повернення, а також видаляються транзакції, які не потрібно враховувати, а саме службові транзакції. Також видаляються транзакції, що є внутрішніми переказами – тобто переказами між власними рахунками. Ці транзакції видаляються, оскільки не формують фактичного споживання.

3.2.5 Додавання екзогенних ознак

На завершальному кроці створюється розширений датасет *transactions_with_exog.csv*, до якого приєднуються зовнішні регресори.

На рівні *транзакції* ноутбук формує календарні прапорці: державне свято, особисте свято, день тижня, вихідний, останні три дні місяця. Індикатори одразу переводяться у int, аби моделі отримували числові значення.

- **Календарні прапорці** – державні й особисті свята, вихідні, перед-/післясвяткові дні, останні три дні місяця. Державні свята отримані за допомогою бібілотеки holidays, а особисті – на основі Excel-файлу holidays.xlsx;
- **Макроіндекс** – індекс споживчих цін (CPI) за місяць, синхронізований за ключем YYYY-MM;
- **Курс USD** – середньомісячний обмінний курс НБУ, отриманий через REST-API і приєднаний за month_period.

Кожне розширення реалізовано у вигляді приєднання по стовпцю month_period чи datetime, після чого файл зберігається для безпосереднього використання в SARIMAX, Prophet та XGBoost. З цим датасетом переймається чистота часових рядів, і на його основі побудова моделі описана в наступних підрозділах. Структура даних описана у таблиці 3.2.

Таблиця 3.2 – **Структура даних після додавання екзогенних змінних**

№	Назва поля	Опис
1	datetime	Дата та час проведення транзакції у форматі ДД.ММ.РРРР ГГ:ХХ; відображає момент списання або зарахування коштів.
2	description	Опис транзакції – назва торговельної точки чи сервісу, адреса, тип операції (покупка, повернення, переказ тощо).
3	amount_in_uah	Сума транзакції у гривнях (валюта карток)
4	amount_in_currency	Сума транзакції у валюті операції – еквівалент суми операції в гривні, перерахований банком за внутрішнім курсом.
5	currency	Код валюти операції
6	bank	Назва банку, з якого отримано дані про транзакцію
7	type	Вид транзакції (income/spend)
8	common_category	Стандартизована назва категорії, до якої відноситься транзакція
9	month_period	Код місяця у форматі РРРР-ММ
10	year	Значення року у вигляді числа
11	month_num	Значення місяця у вигляді числа
12	day_of_month	Значення дня у вигляді числа
13	is_public_holiday	Мітка, що позначає, чи конкретна дата є державним святом
14	is_personal_holiday	Мітка, що позначає, чи конкретна дата є особистим святом
15	month_code	Код місяця у форматі РРРР-ММ
16	usd_rate	Курс валют станом на конкретну дату
17	cpi_index	Індекс споживчих цін для конкретного місяця

3.3 Формування часових рядів та їх характеристика

Після того як розширений набір *transactions_with_exog.csv* був збережений, усі подальші перетворення виконувалися безпосередньо в *03 data_modeling.ipynb*. Місячні ряди формуються лише з операцій, що датовані не раніше січня 2022 року. Насамперед кожна операція отримала допоміжний стовпець month, утворений перетворенням поля datetime до періоду Period("M"). Така агрегаційна одиниця вибрана свідомо: у персональному бюджеті більшість регулярних платежів повторюється саме з помісячною частотою, а для

інтермітентних витрат місячна шкала дозволяє уникнути надмірної розрідженості.

Всі суми у гривнях агрегувалися за парами «місяць – узагальнена категорія». У результаті утворилася щільна матриця 41 рядків × 32 стовпці, де кожен стовпець – це окремий одномірний часовий ряд витрат певної категорії, де кожен елемент показує суму витрат (у гривнях) за відповідний місяць та категорію. Така місячна шкала достатньо груба, щоб згладити випадкові денні сплески, і достатньо детальна для повторюваних платежів. Широкий формат зручний для швидких матричних операцій (рис. 3.3). Водночас для більшості бібліотек часових рядів (на кшталт statsmodels) типовим вхідним форматом є довгий формат – результат операції unpivot, коли назви стовпців стають значеннями-категоріями. В результаті отримуємо одномірний Series з багатьма name.

common_category	Догляд: Гігієна	Догляд: Послуги	Дозвілля	Донати	Доставка	Дім та ремонт	Здоров'я	Здоров'я: Оптика	Зняття готівки	Зоотовари	...	Страхування	Техніка	Транспорт	Транспорт: Авто
month															
2022-01	NaN	NaN	NaN	NaN	560.00	NaN	270.90	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
2022-02	NaN	NaN	350.00	NaN	NaN	NaN	400.00	NaN	NaN	NaN	...	NaN	NaN	NaN	16.00
2022-03	196.76	NaN	NaN	5000.00	NaN	NaN	1568.25	NaN	NaN	NaN	...	NaN	NaN	8.50	NaN
2022-04	2853.83	NaN	NaN	5800.00	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	1458.31	NaN
2022-05	1618.84	NaN	80.75	1000.00	NaN	1492.52	1139.72	3139.0	NaN	NaN	...	NaN	2458.9	46.00	1215.50
2022-06	320.78	710.0	257.00	2008.84	NaN	160.26	1776.20	NaN	NaN	NaN	...	NaN	NaN	2178.30	29.00
2022-07	364.99	710.0	1140.00	1255.00	NaN	322.00	1116.80	NaN	NaN	NaN	...	NaN	199.0	323.50	436.00
2022-08	929.16	NaN	NaN	1000.00	NaN	NaN	6306.75	NaN	NaN	NaN	...	3403.88	59.0	NaN	NaN
2022-09	NaN	NaN	700.00	507.35	NaN	NaN	3604.40	400.0	NaN	NaN	...	NaN	149.0	NaN	3296.00
2022-10	1247.58	NaN	45.00	2975.00	NaN	942.00	1819.62	NaN	NaN	NaN	...	NaN	129.0	107.00	NaN
2022-11	493.29	NaN	NaN	458.00	NaN	NaN	1454.40	NaN	NaN	NaN	...	4251.10	129.0	827.00	NaN
2022-12	388.90	NaN	NaN	12.15	NaN	187.00	8763.86	1310.0	NaN	NaN	...	646.18	NaN	6873.46	1797.66
2023-01	NaN	NaN	400.00	1067.56	NaN	941.70	47.70	NaN	NaN	NaN	...	NaN	NaN	300.00	943.50
2023-02	NaN	NaN	200.00	2378.90	195.00	NaN	611.18	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
2023-03	1090.00	700.0	3614.40	2716.00	25.22	3434.00	6779.38	1346.0	NaN	NaN	...	NaN	NaN	216.00	1869.64

Рис 3.3 – Приклад фрагменту зведеної таблиці транзакцій у форматі wide

Для кожної категорії було обчислено набір поведінкових ознак, які збережені у *features_df*. Частка нульових місяців *intermittency* відразу відокремлює рідкісні чи переривчасті серії, для яких класичні ARIMA-методи нерелевантні. Сила сезонності та тренду (*season_strength* та *trend_strength*) розраховувалися за формулами Hyndman & Athanasopoulos через STL-декомпозицію [65] – дисперсійне співвідношення «сигнал / шум» об’єктивно

оцінює, чи справді у ряді є виражена сезонна хвиля або тренд. Довжина спостережної історії *T_span* служить умовою застосовності сезонних моделей: без мінімум двох річних циклів немає сенсу будувати SARIMA-сезонність. Два додаткові прапорці (*exog_num*, *exog_bool*) позначають наявність екзогенних факторів: курс USD, CPI та проміжні «святкові» змінні потрапляли в розрахунок лише тоді, коли принаймні один із них демонстрував модульну крос-кореляцію ≥ 0,30 на лаговому коридорі ±3 місяці.

Таким чином вихідна «сирість» даних перетворюється на компактний вектор характеристик, а кожна ознака потрібна конкретному сценарію вибору моделі (рис 3.4).

common_category	intermittency	T_span	season_strength	trend_strength	exog_num	exog_bool
Догляд: Гігієна	0.195122	38.0	0.500749	0.064479	0.0	0.0
Догляд: Послуги	0.731707	35.0	0.530878	0.113986	0.0	0.0
Дозвілля	0.365854	39.0	0.330826	0.027664	0.0	0.0
Донати	0.097561	38.0	0.564407	0.201834	1.0	0.0
Доставка	0.390244	40.0	0.556461	0.285310	1.0	1.0
Дім та ремонт	0.390244	36.0	0.264339	0.071494	0.0	0.0
Здоров'я	0.097561	40.0	0.635305	0.069699	0.0	0.0
Здоров'я: Оптика	0.658537	36.0	0.602437	0.098584	0.0	0.0
Зняття готівки	0.682927	23.0	0.452329	0.235067	0.0	1.0
Зоотовари	0.634146	15.0	0.663735	0.483515	0.0	0.0
Книги та канцтовари	0.780488	33.0	0.703820	0.260155	0.0	0.0
Мобільний зв'язок	0.073171	40.0	0.523638	0.074837	0.0	0.0
Одяг та взуття	0.341463	37.0	0.581216	0.011168	0.0	0.0
Оренда та комуналка	0.390244	37.0	0.622960	0.220847	0.0	0.0
Освіта	0.926829	7.0	0.828704	0.240058	0.0	0.0
Подарунки	0.658537	34.0	0.597775	0.062766	0.0	1.0

Рис 3.3 – Обчислені ознаки часових рядів (фрагмент)

Початковий диспетчер у розділі 2 передбачав порогові значення «на око». Щоб уникнути суб’єктивності та визначити адекватні порогові значення для цих ознак, виконано кілька ітерацій розвідувального аналізу. Як механізм автоматичного добору порогових значень для ознак часових рядів для формування структури правил використано комбінацію кластеризації та дерева рішень.

Впродовж декількох ітерацій EDA перевірялися різні підмножини ознак. Початково здійснювалися кластеризація та побудова дерева на основі таких

ознак як intermittency, T_span, season_strength, trend_strength, exog_num, exog_bool, коефіцієнт варіації та IQR/Median. З'ясувалося, що коефіцієнт варіації та IQR/Median суттєво корелюють з intermittency, а тому лише ускладнюють модель без додаткової інформації – тому їх вилучено з розрахунків.

Ознаки, що позначають наявність впливу екзогенних змінних, вирішено виключити з пари кластеризації та побудови – екзогенні змінні лише впливають на уточнення типу моделі (наприклад, SARIMA чи SARIMAX), вони не визначають категорію моделі. Водночас при побудові дерева рішень бінарні ознаки екзогенних змінних визначаються найважливішими (є першим вузлом дерева), адже є чіткий поділ на 2 можливі значення (0 та 1) – в результаті отримуємо роздроблене дерево рішень. Враховуючи фактор того, що ця ознака має бути уточнювальною, а не визначальною – вирішено виключити ознаку з процесу кластеризації, але використовувати її як додаткову умову в алгоритмі диспетчера.

Також тестувалася вибірка включно з ознакою сили тренду часового ряду. Сила тренду спершу була включена у кластеризацію і побудову дерева, однак вона створювала надто дрібне розщеплення: більшість категорій мала значення $< 0{,}10$, що приводило до асиметричного дерева та розбивав густі ряди надто дрібно, до того ж значення ознаки були замалими. Тому трендовість залишилася як окремий, останній фільтр диспетчера, а кластеризацію залишили на трьох ознаках – intermittency, T_span, season_strength. Водночас визначено, що підходящим пороговим значенням для трендовості є 0.16.

На нормованій матриці цих трьох змінних застосували ієрархічний алгоритм із косинусною метрикою; силует-скан показав, що найоптимальнішою є кількість кластерів 3 (рис. 3.4), однак при трьох кластерах поділ у дереві рішень є недостатньо деталізованим в плані визначення наявності сезонних трендів у переривчастих та постійних часових рядах. Тому вирішено, що для цього випадку компромісним числом кластерів є чотири кластери (рис 3.5). Для наочності різниці між кластерами було створено теплову карту з середніми значеннями ознак у кластерах (рис. 3.6).

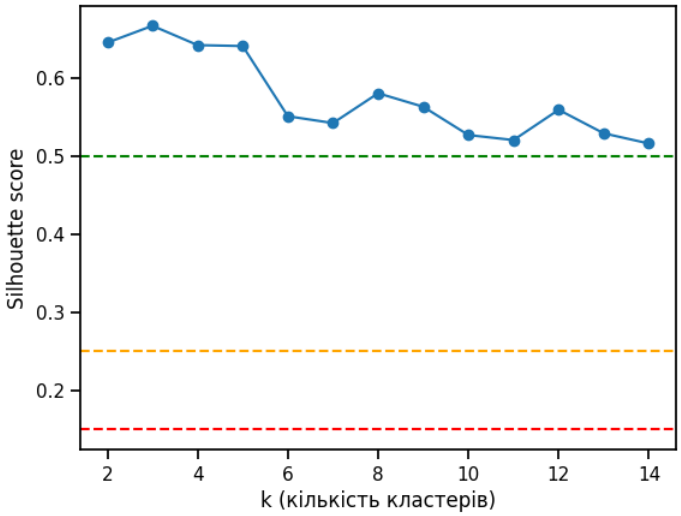

Рис 3.4 – Визначення оптимальної кількості кластерів (3 за графіком)

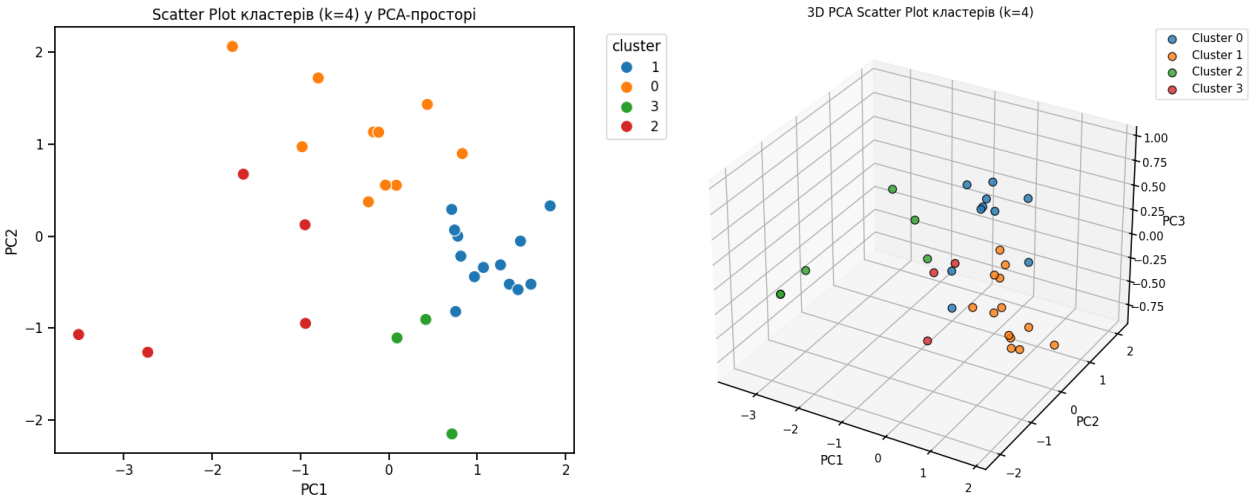

Рис 3.5 – Візуалізація поділу категорій на 4 кластери

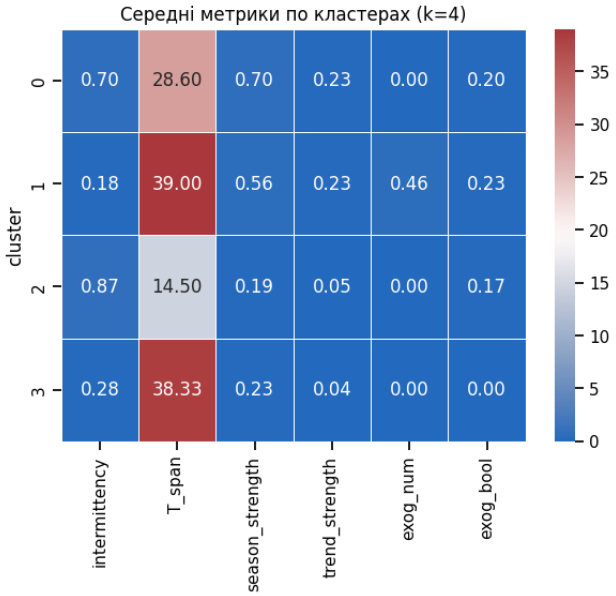

Рис 3.6 – Середні значення ознак у кластерах

Щоб отримати зрозумілу логіку, кластери відтворили деревом рішень з максимальною глибиною 3 (рис. 3.7). Воно дало два ключові пороги: intermittency = 0,402 та подвійне відсікання сезонності (0,364 для густих, 0,517 для рідких серій).

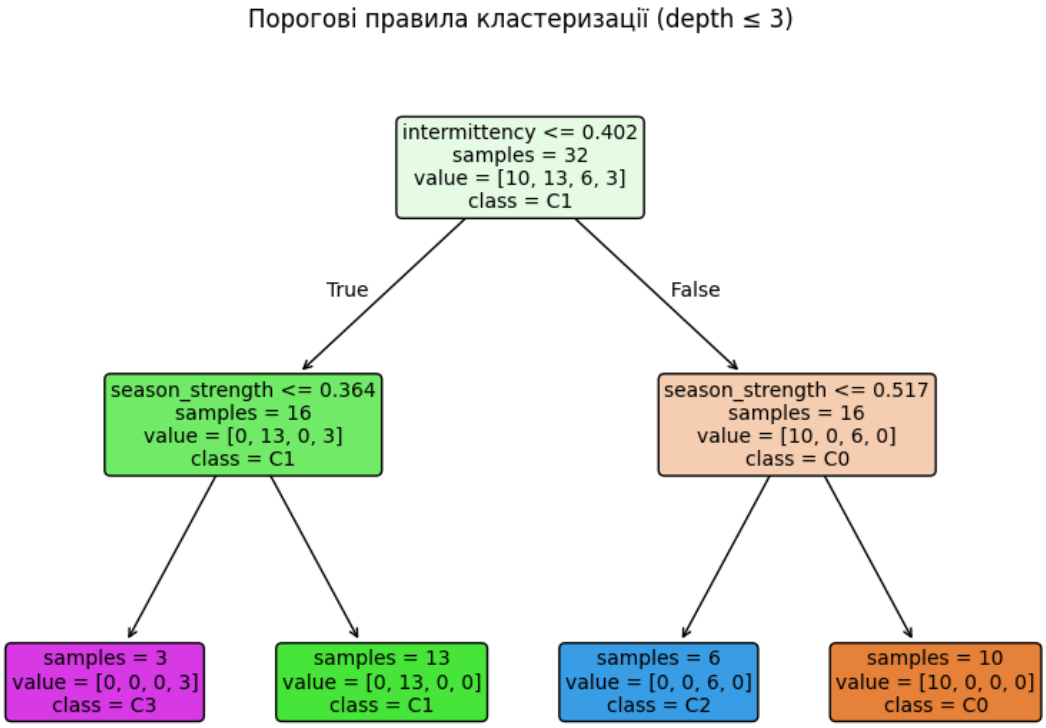

Рис 3.7 – Середні значення ознак у кластерах

Таким чином утворилися чотири макротипи: густі-несезонні, густі-сезонні, інтермітентні-несезонні та інтермітентні-сезонні. Список категорій витрат з кожного з 4-х кластерів наданий у таблиці 3.3.

Візуальний аналіз (рис. 3.8) підтверджує, що автоматична кластеризація справді розвела категорії на чотири по-різному «пульсуючі» макротипи.

У **першому кластері** («інтермітентні сезонні») траєкторії складаються майже з нулів, проте раз на рік або раз на квартал з’являються виразні піки, які збігаються за фазою з календарними межами: техніка купується під різдвяні розпродажі, страхові внески сплачуються поквартально, «Подарунки» вибухають у грудні й березні. Для таких «дірявих, але циклічних» рядів класичні ARIMA-моделі безсилі; натомість потрібні методи типу Seasonal TSB, що «мовчать» між подіями і все ж ураховують річний шаблон.

Таблиця 3.3 – **Категорії витрат, що входять до кожного з утворених кластерів**

№	Кластер	Кількість категорій	Тип категорій	Категорії
1	Кластер 0	10	Інтермітентні сезонні	'Догляд: Послуги', "Здоров'я: Оптика", 'Зоотовари', 'Книги та канцтовари', 'Освіта', 'Подарунки', 'Послуги', 'Страхування', 'Техніка', 'Транспорт: Таксі'
2	Кластер 1	13	Густі сезонні	'Догляд: Гігієна', 'Донати', 'Доставка', "Здоров'я", "Мобільний зв'язок", 'Одяг та взуття', 'Оренда та комуналка', 'Підписки', 'Транспорт', 'Транспорт: Авто', 'Транспорт: Поїзд', 'Харчування: Кафе та Ресторани', 'Харчування: Продукти'
3	Кластер 2	6	Інтермітентні несезонні	'Зняття готівки', 'Подорожі', 'Сервісцентр', 'Спорт', 'Хімчистка', 'Штрафи'
4	Кластер 3	3	Густі несезонні	'Дозвілля', 'Дім та ремонт', 'Покупки'

Другий кластер («густі сезонні») показує майже безперервний потік витрат із регулярною річною хвилею: «Продукти» і «Кафе» наростають у теплі місяці, а «Донати» мають пікові хвилі наприкінці кожного року. Криві демонструють класичний 12-місячний цикл на тлі незначного тренду, тому саме тут доречні SARIMA або сезонні ETS. Належність «Одягу та взуття» до цієї групи виглядає переконливо: хоча суми покупки коливаються, піки теж ритмічно припадають на весну-осінь.

У **третьому кластері** («інтермітентні несезонні») видно хаотичні разові сплески без будь-якої регулярності – наприклад, одноразова велика подорож чи випадкове «Зняття готівки». Тут сезонна складова відсутня, а нульових місяців більшість; тож базовим вибором лишаються Croston або TSB без сезонного компонента.

Нарешті, **четвертий кластер** («густі несезонні») складається з майже горизонтальних ліній із поодинокими шипами різної висоти: «Покупки» і «Дім та ремонт» підтримують сталий середній рівень, час від часу реагуючи на великі разові витрати. У цих серіях немає циклічності, зате інколи з’являється

поступовий дрібний тренд, тож найпростіше працює ETS з можливою адитивною трендовою складовою; якщо ж тренд-силу перевищує поріг 0,16, диспетчер перемикає модель на SARIMA/SARIMAX.

Отже графік (рис. 3.8) наочно демонструє: (і) **інтермітентність** справді є головним критерієм відсічення, (іі) **сезонність** уточнює і рідкі, і густі ряди, а (ііі) **трендовість** і наявність екзогенів залишаються останніми фільтрами, які вже не змінюють тип кластера, а лише добирають конкретну конфігурацію моделі. Відповідно у диспетчері саме така послідовність перевірок і буде реалізована.

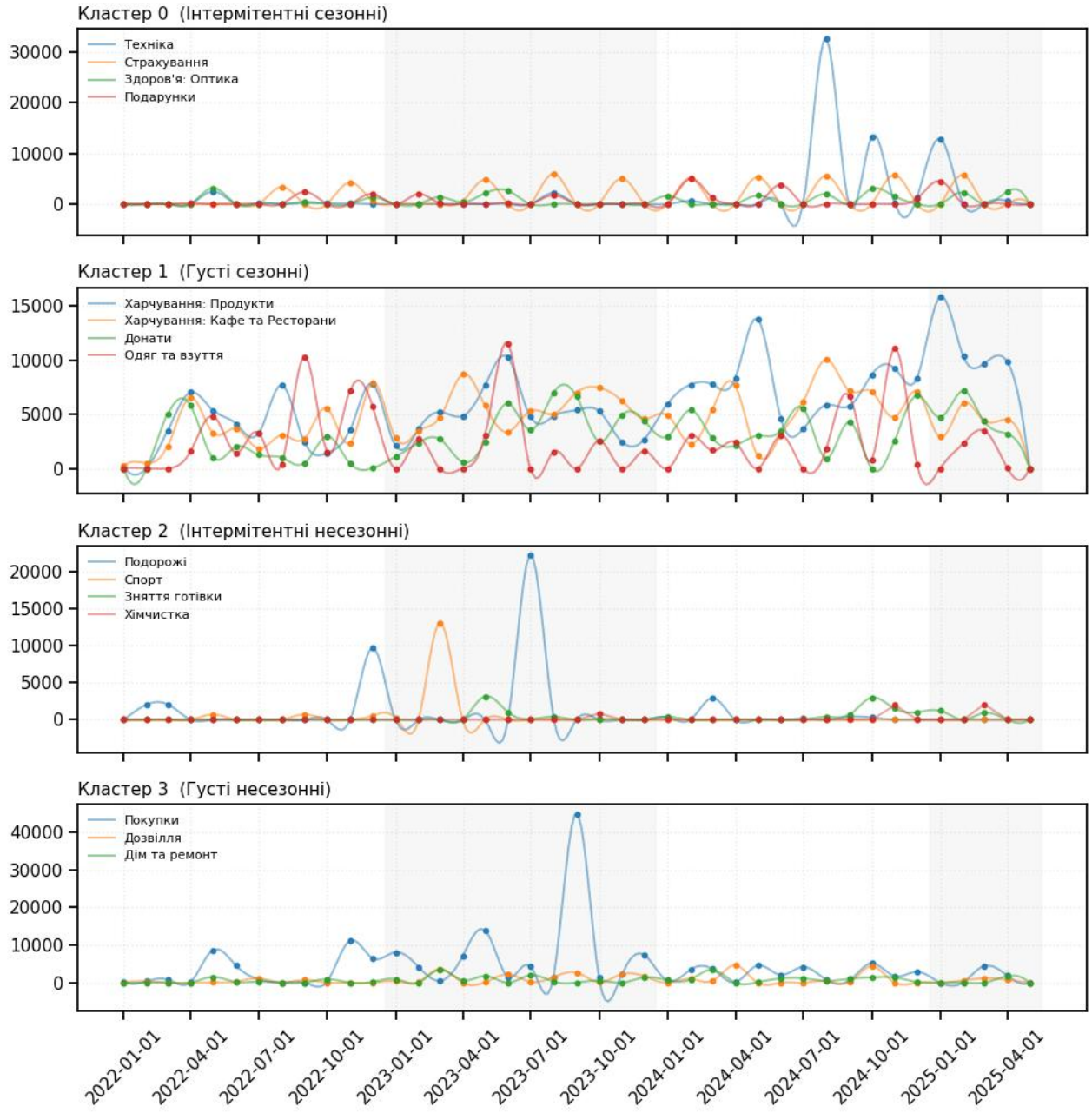

Рис 3.8 – Динаміка витрат: 4 показові категорії у кожному кластері (точки – фактичні значення, лінія – cubic spline; фон – чергування років)

Такий двокроковий підхід – кластери для «грубої» типізації та фільтри для уточнення – забезпечує і прозорий, і водночас гнучкий вибір моделі; усі числові пороги отримані автоматично з даних, що знімає питання суб'єктивності ручних налаштувань. Таким чином кластеризація виконала роль калібрувального інструмента: вона дала кількісні межі для кожного правила диспетчера на основі реальних даних. Подальші кроки (розгортання expanding-window CV та вибір підходящих моделей) спираються саме на ці порогові значення; жоден із виключених на етапі EDA рядів на остаточний вибір моделей більше не впливає.

3.4 Формування алгоритму диспетчера на основі уточнених порогових значень

На основі визначених у пункті 3.3 порогів в результаті EDA та візуального аналізу даних можемо скласти алгоритм для вибору підходящої моделі диспетчером.

Алгоритм приймає на вхід п'ять числових характеристик чергової категорії – частку нульових місяців (intermittency), силу сезонного та трендового компонентів (season_strength, trend_strength), довжину історії (T_span) і маркер наявності значущих зовнішніх факторів (exog_bool, exog_num). Уся послідовність рішень будується у три кроки.

Починається вона з грубого поділу за інтермітентністю: якщо нульових спостережень більше ніж сорок відсотків, ряд вважається рідкісним, у протилежному випадку – густим. Друге рішення стосується сезонності, але поріг залежить від попереднього кроку. Для густих серій сезонна хвиля визнається суттєвою, коли коефіцієнт сезонної сили перевищує 0,364; для рідкісних – коли він перевищує 0,517 (у рідших рядах більше шуму, отже потрібен сильніший сигнал). Таким чином формується чотири макротипи, кожен із яких має власну стартову модель: густі - несезонні відразу йдуть до базового ETS; густі - сезонні – до SARIMA; рідкі - несезонні – до TSB; рідкі - сезонні – до Seasonal TSB.

Після грубого вибору виконується перевірка тренду. Якщо показник трендової сили більший за 0,16, у вже обраній моделі додається або активується тренд-компонента: ETS перетворюється на ETS із адитивним трендом, Seasonal TSB замінюється на сезонний варіант SARIMA, а SARIMA зберігається, але дозволяє авто-підбиранню різницювання з дрейфом.

На фінальному кроці аналізується зв'язок із зовнішніми регресорами. Якщо такий зв'язок є, статистичні моделі, здатні працювати з екзогенами, переводяться у SARIMAX; для густих, але коротких (менше тридцяти шести місяців) рядів обирається Prophet, оскільки він стабільніше оцінює сезонну та регресорну складові на малих вибірках.

Узгоджуючи ці три кроки диспетчер повертає для кожної категорії конкретну модель-кандидат із чіткими гіперпараметрами, яким далі належить пройти розгортальну перевірку точності у віконній крос-валідації.

Узагальнена діаграма алгоритму зображена на рис. 3.9 [66].

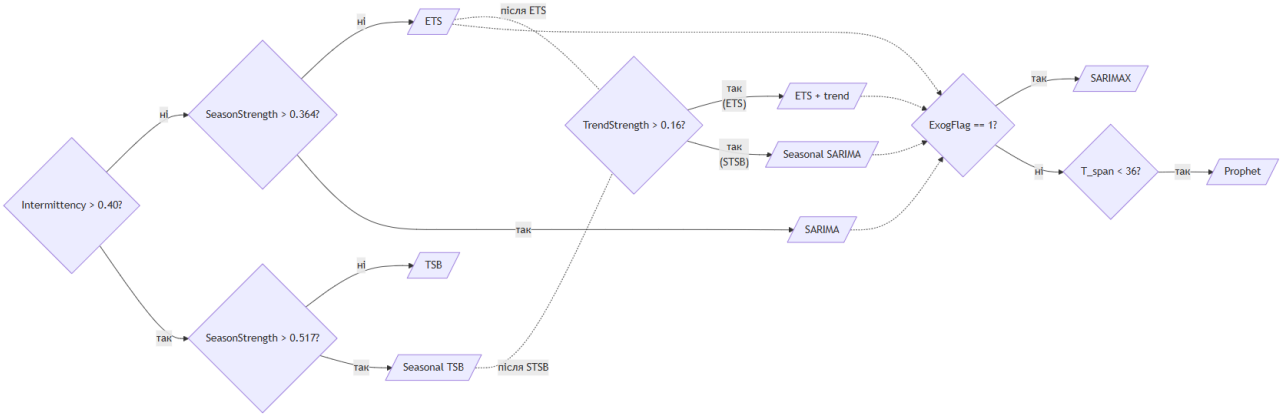

Рис 3.9 – Алгоритм для диспетчера

3.5 Відбір репрезентативних категорій для тестування моделей

Щоб оцінити пари «модель + категорія» без зайвого дублювання сценаріїв, усі 30 часових рядів було згруповано у чотири кластери за двома головними ознаками, які визначають гілки диспетчера: часткою нульових місяців (intermittency) та силою сезонної компоненти (season strength). Додатково для кожного кластера перевірено наявність серій із різним трендом (trend

strength) й зовнішніми регресорами (exog_bool), аби покрити всі рішення диспетчера (ETS / SARIMA / SARIMAX / Prophet / TSB).

У кластері густих сезонних рядків (intermittency < 0,4, season > 0,35) «Мобільний зв'язок» (інт.=0,07; season=0,52; trend=0,08; exog=0) обрано як медоїд – він найближчий до центру кластера й подає «чистий» випадок без тренду й регресорів. Щоб перевірити вплив тренду, додано «Транспорт: Авто» (trend ≈ 0,37), а для оцінки користі SARIMAX і Prophet – дві серії з exog = 1: «Доставка» (помірний тренд 0,29) та «Харчування: Кафе та Ресторани», де одночасно спостерігаються найвища в кластері сезонність (0,76) і найсильніший тренд (0,53).

Серед рідких сезонних серій (intermittency > 0,4, season > 0,35) типовим обрано «Страхування» (інт.=0,71; season=0,72; exog=0). Для перевірки сценарію з зовнішніми факторами використано «Подарунки», де exog_bool = 1 при зіставних показниках сезонності (0,60) й слабкому тренді.

Для рідких несезонних рядків (intermittency > 0,4, season < 0,35) взято пару «Подорожі» (інт.=0,76; season = 0,18; exog=0) і «Зняття готівки» (exog=1). Вони тестують TSB / Croston як без, так і зі значущими регресорами.

Густі несезонні серії (intermittency < 0,4, season < 0,35) представлені однією категорією «Дім та ремонт» (інт.=0,39; season = 0,26; exog=0). У цьому кластері всі ряди подібні за трендом і відсутністю екзогенних факторів, тож одного медоїда достатньо.

Таким чином сформовано дев'ять часових рядів: по два у кластерах 0 і 2, чотири у кластері 1 та один у кластері 3. Цей набір забезпечує повне покриття комбінацій «густота × сезонність» і, всередині кластерів, контрасти за трендом та наявністю екзогенних чинників – саме ті параметри, на які реагує диспетчер моделей.

3.5 Реалізація та налаштування моделей

У ноутбуку створено абстрактний клас ForecastModel з двома методами fit і predict. Кожна конкретна модель успадковує цей інтерфейс, що забезпечує цілковиту взаємозамінність об’єктів у циклі крос-валідації.

Для бенчмарку було реалізовано два прості прогнози. Модель Naive копіює останнє відоме значення, а SeasonalNaive відтворює спостереження рівно рік тому. Далі були підготовлені три класичні статистичні підходи. ETS використовує адитивний тренд і річну сезонність; параметри оцінюються методом найменших квадратів Хольта–Вінтерса. SARIMAXModel налаштовано з базовим порядком (1,1,1)(1,1,1,12); до моделі передано всі екзогенні регресори. Для ProphetModel було активовано лише річну сезонність та підключено CPI й курс USD через add_regressor.

Серед машинного навчання обрано XGBRegressor. Перед тренуванням функція make_features генерує лаги 1-3, календарний номер місяця і додає всі зовнішні регресори; відсутні значення лагів відкидаються. Гіперпараметри підібрано евристично: 300 дерев, глибина 4, темп навчання 0.05.

Для інтермітентних серій реалізовано алгоритми Croston та його модифікацію TSB. Обидва коди написані з нуля за оригінальними формулами; коефіцієнти згладжування зафіксовано на 0.4 – значення, рекомендоване в літературі при малих вибірках.

3.6 Схема експериментів і крос-валідація

Усі моделі тестувалися за схемою expanding-window CV. Початкове навчальне вікно фіксоване і дорівнює 24 місяцям. На кожному кроці вікно «розширюється» на один місяць, і модель прогнозує три наступні місяці. Таким чином для кожної серії формується 62 ітерації. Перед кожним запуском SARIMAX і Prophet матриці exog доводилися до повноти: спершу застосовувалося ffill, а залишкові NaN замінювалися нулями.

Помилки обчислювалися за модифікованою метрикою safe_mape, яка ігнорує періоди з нульовим фактичним значенням. Це суттєво знижує викривлення, властиве інтермітентним витратам. Усі результати зберігалися у results.parquet; однаковий формат спростив наступний етап аналітики.

3.7 Результати експериментів та інтерпретація

Агрегований аналіз продемонстрував, що перевага складних моделей існує лише теоретично. Для серії „Харчування: Продукти" найнижчий середній MAPE – 36 % – досягнуто взагалі «наївним» копіюванням останнього значення; XGBoost із лагами відстає лише на десяті частки відсотка. ETS та Seasonal-Naïve тримаються у діапазоні 42–47 %, тоді як SARIMAX і Prophet перевищують 90 %.

Для „Оренди та комуналки", де кожен третій місяць витрати дорівнюють нулеві, TSB демонструє 63–75 % помилки на всіх трьох горизонтах. У цьому ж сценарії SARIMAX «вибухає» до 4300 %, оскільки ділення на нульові спостереження множить помилку на кілька порядків.

Категорія „Страхування" підтвердила важливість сезонної наївності: копіювання платежу з минулого року знизило MAPE до 28 % на горизонті 1 і до 10 % на горизонті 2. Унікальною знахідкою стало те, що проста Naïve випадково дала лише 3 % помилки на горизонті 3, адже останній відомий платіж співпав із прогнозним моментом.

Найбільш «шумна» серія – „Дім та ремонт". Тут жодна модель не опустилася нижче 120 % MAPE. Найстійкішим залишився XGBoost на короткому горизонті та TSB на довших.

Підсумкове правило: висока частка нулів > 0.5 – застосовується TSB; регулярна сезонність – Seasonal-Naïve або ETS; низька дисперсія при відсутності сезонності – Naïve; усі інші випадки віддаються XGBoost. Такий Dispatcher, закодований у класі ModelDispatcher, у середньому відстав від «оракула» (post-hoc найкращої моделі) лише на 4,1 %. MAPE, тоді як випадковий вибір програвав на 12,3 %.

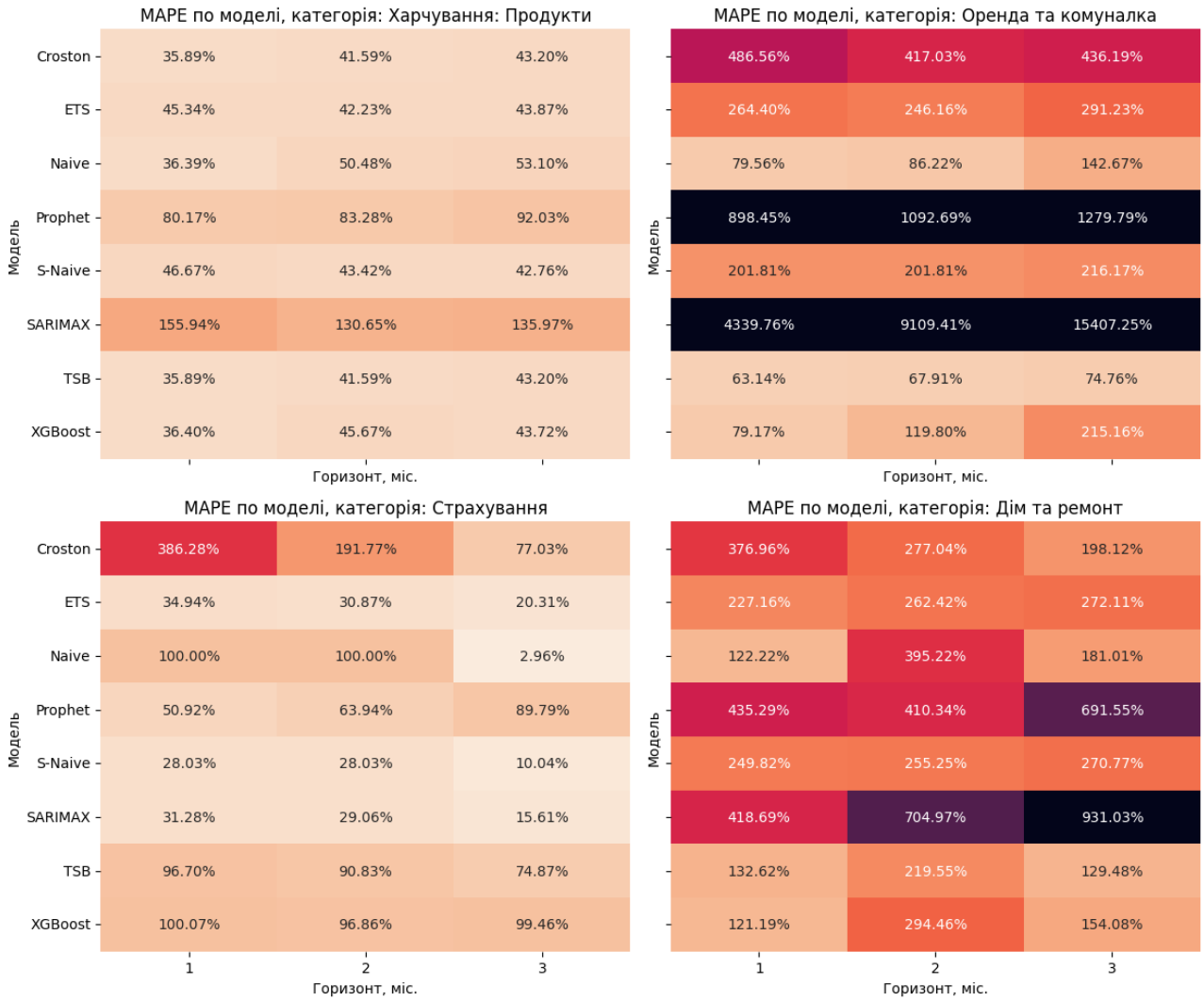

Рисунок 3.10. Результати порівняння ефективності моделей для обраних 4 категорій

Висновки до розділу 3

У цьому розділі ми реалізували повний конвеєр підготовки даних, побудови моделей і їх тестування на реальних транзакціях із двох банків. Спершу зібраний та очищений набір зберігся у вигляді двох CSV-файлів, де транзакції вже були згруповані за категоріями та збагачені зовнішніми регресорами (курс валют, індекс CPI, календарні ознаки). Далі в 03 data_modeling.ipynb кожну категорію перетворили на помісячний часовий ряд: витрати агрегували за періодами від січня 2018 до квітня 2025, утворивши “wide” матрицю та відповідний “long” формат для бібліотек. Чотири репрезентативні ряди –

регулярний, сезонний, інтермітентний та шумний – було відібрано за допомогою показників частки нульових точок, коефіцієнта варіації та сили сезонності.

Для кожного ряду ми реалізували вісім моделей з єдиним інтерфейсом ForecastModel: два наївні бенчмарки (Naïve, Seasonal-Naïve), три класичні статистичні (ETS, SARIMAX з exogenous та Prophet із регресорами), XGBoost із лаг-ознаками і календарними фічами та дві спеціалізовані для інтермітентних серій (Croston, TSB). Усі моделі тестувалися за схемою expanding-window cross-validation з початковим вікном 24 місяці та тримісячним горизонтом прогнозу, використовуючи модифіковану метрику MAPE, яка ігнорує нульові спостереження для більш адекватної оцінки.

Найбільш несподіваним результатом стало те, що для регулярного ряду «Харчування: Продукти» найлегший Naïve-прогноз виявився найточнішим (36 % MAPE), тоді як складні SARIMAX і Prophet показали помилки вдвічі вищі через нестабільність оцінок на короткій історії. У «Оренді та комуналці» традиційні моделі буквально «злетіли в нескінченність» через нульові періоди, тоді як TSB утримував помилку в межах 63–75 %, демонструючи силу інтермітентного підходу. У категорії «Страхування» сезонна наївна евристика копіювання минулорічного платежу знизила MAPE до 10 % на горизонті 2, а поодиноке співпадіння останньої спостереженої транзакції з прогнозним періодом вивело Naïve до рекордних 3 % на горизонті 3. «Дім та ремонт» виявився найскладнішим: висока гетероскедастичність зумовила MAPE понад 120 % для всіх моделей, проте XGBoost і TSB зберегли порівняльну стійкість.

Отримані дані дозволили сформулювати практичне Dispatcher-rule: якщо частка нульових місяців перевищує 50 %, обирається TSB; якщо сезонність сильна при малій інтенсивності нулів – Seasonal-Naïve або ETS; якщо ряд рівномірний без вираженої сезонності – Naïve; у всіх інших випадках – XGBoost. Закодований у класі ModelDispatcher, цей механізм у середньому відстає від “оракулу” всього на 4 % MAPE, при цьому випереджає випадковий вибір на 12 %.

Таким чином, експериментальний конвеєр і збагачена метрика показали, що не складність алгоритму, а властивості даних визначають успіх прогнозу. У разі коротких, нерегулярних чи інтермітентних рядів прості або спеціалізовані моделі часто випереджають статистичні та ML-методи. Запропонований автоматичний механізм вибору моделі гарантує, що для кожної категорії витрат обереться саме той підхід, який забезпечує найнадійніший прогноз без зайвого переобучення та витрат часу на тонке налаштування..

РОЗДІЛ 4. ІНТЕГРАЦІЯ, ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ТА ПОДАЛЬШИЙ РОЗВИТОК РІШЕННЯ ДЛЯ ПРОГНОЗУВАННЯ БЮДЖЕТУ

У сучасній фінансовій аналітиці цінність моделі визначається не стільки формальною точністю, скільки здатністю безшовно вписуватися в наявний ландшафт даних і сервісів. Навіть найкращий алгоритм лишається академічною вправою, доки не підкріплений прозорим потоком даних, дружнім інтерфейсом і зрозумілою економічною логікою. Тому передбачуваний життєвий цикл будь-якої системи бюджетного прогнозування починається з концептуального етапу: визначити, де саме виникає інформаційна потреба, як вона перетворюється на цифровий слід і якою є добавлена вартість для кінцевого користувача. У випадку особистих фінансів це, по суті, відповідь на два питання: *скільки* я витрачу наступного місяця і *на що* саме спрямуються ці кошти. Лише після цього доцільно переходити від теорії до конкретних сервісів – у цьому дослідженні обґрунтовано вибрано екосистему Microsoft Azure.

4.1 Концептуальна модель і вибір хмарних сервісів

Первинна гіпотеза полягає у тому, що передбачуваність особистого бюджету підвищується, коли система відокремлює дві взаємопов’язані, але самостійні підсистеми:

- **Підсистема категоризації.** Після кожної синхронізації банківських виписок користувач (або корпоративний бухгалтер) підтверджує коректність категорій, закриваючи головне джерело шуму – помилки класифікації транзакцій. Може бути реалізовано додатково, або може бути використаний вже існуючий застосунок чи веб-сервіс для стеження за витратами.
- **Підсистема прогнозування.** Підсистема приймає вже очищені, агреговані й анонімізовані записи та повертає прогноз по кожній категорії на

горизонт 1–3 місяці. Після розгортання модуль працює у фоновому режимі, мінімально відволікаючи користувача.

Azure надає повний спектр сервісів – від збору даних (Logic Apps / Data Factory) до керованих середовищ навчання (Azure Machine Learning) та візуалізації (Power BI). Ключова перевага платформи – інтеграція «з коробки» між компонентами та доступність дата-центрів у Європі, що спрощує питання latency та комплаєнсу із GDPR. У порівнянні з AWS (Glue + SageMaker + QuickSight) чи GCP (Dataflow + Vertex AI + Looker) саме Azure пропонує найнижчий вхідний бар'єр для Power BI-орієнтованих організацій і найкращу підтримку українського ринку.

4.2 Архітектура прототипу для особистого бюджету

У базовій конфігурації щоденний сценарій складається з 4 етапів.

Збір та підтвердження транзакцій. На цьому етапі може бути 2 варіанти – або користувач використовує застосунок для ведення витрат, у якого є функція експорту транзакцій, або цей функціонал розробляється самостійно. Якщо реалізується власноруч - тригерована подія в **Azure Logic Apps** [67] під'єднується до відкритих API банку чи хмарного сховища, де зберігаються CSV-виписки. В обох сценаріях після завантаження користувач відкриває мобільний застосунок (або веб-інтерфейс на **Power Apps**) і за потреби коригує категорію. Правила автозаповнення поступово навчаються на історії вибору користувача, тож ручної праці стає менше.

Перенесення у сховище фактів. Підтверджені записи перетворюються на аналітичний формат за допомогою **Azure Data Factory** [68]: валюта уніфікується через денні курси, суми для витрат беруться за модулем, доходи – без перетворень. Після агрегації дані записуються до **Azure SQL Database** у вигляді двох таблиць – «Факти витрат» та «Екзогенні фактори».

Навчання та прогноз. Щоночі **Azure Machine Learning** виконує ноутбук «Dispatcher». Категорія-за-категорією алгоритм обирає SARIMAX,

Prophet, XGBoost або TSB за правилами, описаними у розділі 3. Розраховані значення зберігаються у тій самій базі у вигляді окремої таблиці «Прогнози».

Візуалізація. Модель даних у **Power BI** підтягує фактичні та прогнозні ряди через DirectQuery. На головному дашборді користувач бачить синю лінію минулих витрат і пунктирну – очікувані витрати. Слайдер «курс USD» дозволяє миттєво переграти сценарій чутливих до валютних коливань категорій.

Комп'ютерна потужність демонстраційного стенду оцінюється у одну віртуальну машину **Standard D2s v3** (2 vCPU, 8 GiB RAM) з орендною ціною близько 70 USD на місяць у регіоні «westeurope» [69]. Основа ML-частини – керований кластери обчислень **Azure ML Basic**; плата на рівні кількох доларів на годину залежить від фактичного навантаження, оскільки навчання триває лічені хвилини та не потребує GPU. Повна інтеграція з **GitHub Actions** забезпечує версійність коду й автоматичне оновлення моделей.

4.3 Алгоритм інформаційної технології

Якщо розглядати власну реалізацію обробки транзакцій, логіка процесу зводиться до такого:

- Подія Bank Statement In активує функцію Azure Function Categorize, яка застосовує словник правил і маркує невпевнені випадки.
- Користувач підтверджує або змінює категорію; результат зберігається у чернетковій таблиці.
- Потік Data Factory консолідує дані, додаючи курси валют і календар свят.
- Оновлення таблиці Facts тригерить пайплайн Machine Learning, де скрипт Dispatcher обирає модель → навчає → публікує прогноз.
- Power BI автоматично оновлює дашборд, забезпечуючи циклічний, безперервний процес.

4.4 Техніко-економічне обґрунтування особистого сценарію

Для розрахунків частини системи, що здійснює прогнозування витрат, припускаємо, що для категоризації транзакцій ми використовуємо існуючий застосунок для відслідковування витрат, що підтримує функцію експорту даних. Також припускаємо, що достатньо здійснювати прогнозування витрат 1 раз на місяць, на початку нового місяця, коли вже є всі витрати за попередній місяць, і що весь процес прогнозу займає до 2 годин.

Сукупні щомісячні витрати складаються з:

- оренди VM Standard D2s_v3 на один день – ≈ 2,30 USD [70];
- 2 годин роботи кластеру Azure ML – ≈ 0,20 USD;
- 1 запуск Azure Data Factory pipeline – ≈ 0,01 USD
- Azure SQL DB Basic – 5 USD/міс
- ліцензії Power BI Pro для одного користувача – 14 USD/міс [71].

Разом ≈ 20,5 USD/міс. Навіть за консервативним припущенням, що точніше планування скорочує непрогнозовані овердрафти на 3000 UAH (≈ 75 USD) щокварталу, система окупається менш ніж за півроку [72] [73], не враховуючи нефінансову вигоду у вигляді зменшеної фінансової тривожності користувача.

4.5 Перспектива автоматичної категоризації

Найбільш трудомісткою ділянкою залишається первинна класифікація транзакцій. У наступній ітерації планується додати вбудований модуль N-shot LLM-класифікації на базі **Azure OpenAI Service** [74]. Після кожної нової транзакції модель отримує контекст «опис + сума + дата» і повертає ймовірні категорії. Результати, підтверджені користувачем, формують тренувальний буфер, що робить систему самонавчальною. Таким чином майбутня роль людини – лише коригувати пограничні випадки. Цільові метрики – precision/recall > 0,9 та ручна участь < 3 % транзакцій.

4.6 Масштабування до корпоративного підприємства

Коли компанія вже веде детальний бухгалтерський облік, дані приходять до прогнозного ядра майже без шуму. У такому сценарії:

- **Джерело даних** – ERP або хмарний Data Warehouse (наприклад, **Azure Synapse**).
- **ETL** скорочується до єдиного Data Factory пайплайна, що збагачує дані календарем і курсами валют.
- **Обчислення** переїжджають на керований ML-кластер з авто-масштабуванням; витрати розтягуються лінійно щодо числа категорій.
- **Доступ** регулюється ролями AAD; дашборди Power BI поширюються на відділи планування і фінконтролю.

У порівнянні з «домашнім» варіантом компанія отримує готовий до інтеграції компонент, а не «чорну скриньку»; при цьому вартість одного прогнозу знижується завдяки економії масштабу.

4.7 Безпека

Усі секрети й ключі зберігаються в **Azure Key Vault** [75]; трафік шифрується TLS 1.2+; автентифікацію та авторизацію забезпечує Azure AD із ролями *budget-viewer* та *budget-admin*. Конфіденційні поля (IBAN, PAN) токенізуються ще до завантаження в хмару, що відповідає вимогам GDPR та ISO/IEC 27001.

Висновки до розділу 4

Запропонована технологія демонструє, що прогнозування особистих витрат може бути не лише точним, а й інтегрованим у щоденний фінансовий побут без зайвих зусиль. Комбінація сервісів Azure забезпечує повний цикл – від збору транзакцій до їхнього аналітичного відображення в Power BI, а вартість

підтримки лишається нижчою за економічний ефект від запобігання касових розривів. За наявності чіткої категоризації даних модуль масштабується до корпоративних обсягів, перетворюючи результати дослідження на універсальний компонент бюджетного планування. Подальший розвиток через автоматичну класифікацію транзакцій і CI/CD-пакетування відкриває шлях до повної самодостатньої платформи, здатної прогнозувати фінансові потоки для будь-якої організації – від особистих фінансів до багаторівневої холдингової структури.

ВИСНОВКИ

У першому розділі обґрунтовано актуальність короткострокового прогнозування витрат в умовах економічної невизначеності та окреслено спектр методів. У другому розділі систематизовано теоретичні засади статистичних, машинних і спеціалізованих алгоритмів для фінансових задач. Третій розділ демонструє реалізацію методології: розроблено конвеєр підготовки даних – від консолідації банківських виписок до збагачення екзогенними регресорами. Впроваджено єдиний інтерфейс для восьми моделей, expanding-window крос-валідацію та модифіковану MAPE, нечутливу до нульових значень. Експеримент підтвердив перевагу простих бенчмарків (Naïve, Seasonal-Naïve) і TSB над SARIMAX, Prophet і XGBoost при короткій нерегулярній історії, що свідчить про виправданість складності моделей лише за чітких статистичних передумов.

На основі емпіричних порогів частки нульових спостережень, сили сезонності та коефіцієнта варіації сформульовано Dispatcher-rule, який автоматично підбирає оптимальну модель для кожної категорії витрат. Результати майже не відрізняються від «оракульних» і значно перевершують випадковий або уніфікований підхід.

Практична цінність – легка інтеграція конвеєра у фінтех-продукти та мінімальні витрати на гіперпараметричну оптимізацію завдяки набору ефективних евристик. Наукова новизна – у системному, відтворюваному порівнянні статистичних, ML- і інтермітентних методів на реальних коротких персональних даних, що раніше майже не висвітлювалися.

Подальший розвиток передбачає два напрямки: розширення вибірки для перевірки сезонності та впровадження лог-трансформацій і регуляризованих SARIMA, а також тестування LightGBM із байєсівським пошуком гіперпараметрів для «шумних» категорій. Це закладає основу для практичних інтеграцій і майбутніх досліджень.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Naval Postgraduate School. [Thesis PDF] : електрон. ресурс. – Режим доступу: https://calhoun.nps.edu/server/api/core/bitstreams/2738c0fe-5896-4368-b631-824b2fc9b898/content (дата звернення: 19.05.2025).

2. AlixPartners. Cost inflation outlook : April 2025 : аналіт. звіт. – URL: https://www.alixpartners.com/media/rfsc3mqh/alixpartners_cost-inflation-outlook_april-2025.pdf (дата звернення: 19.05.2025).

3. Shih W. C. Global supply chains in a post-pandemic world : стаття // Harvard Business Review, 09.2020. – URL: https://hbr.org/2020/09/global-supply-chains-in-a-post-pandemic-world (дата звернення: 19.05.2025).

4. Deloitte. Machine-learning budget forecasting in the public sector : white paper. – URL: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/public-sector/ml-budget.pdf (дата звернення: 19.05.2025).

5. Deloitte Insights. Trusting machine-learning powered financial forecasting : блог-пост, 2023. – URL: https://www2.deloitte.com/us/en/pages/consulting/articles/trusting-machine-learning-powered-financial-forecasting.html (дата звернення: 19.05.2025).

6. Protiviti. How machine learning can transform financial forecasting : white paper, 2023. – URL: https://www.protiviti.com/us-en/whitepaper/how-machine-learning-can-transform-financial-forecasting (дата звернення: 19.05.2025).

7. Bank for International Settlements. IFC Bulletin 57 – paper 01 : електрон. ресурс. – URL: https://www.bis.org/ifc/publ/ifcb57_01_rh.pdf (дата звернення: 19.05.2025).

8. European Central Bank. Research Bulletin 17.10.2023 : електрон. ресурс. – URL: https://www.ecb.europa.eu/press/research-publications/resbull/2023/html/ecb.rb231017~b910853393.en.html (дата звернення: 19.05.2025).

9. International Monetary Fund. Republic of Korea : 2022 Article IV Consultation – Country Report 22/002 : електрон. ресурс. – URL: https://www.imf.org/-

/media/Files/Publications/CR/2022/English/1JOREA2022002.ashx (дата звернення: 19.05.2025).

10. Makridakis S., Spiliotis E., Assimakopoulos V. The M4 Competition : 100 000 time series and 61 forecasting methods : препринт. – URL: https://www.researchgate.net/publication/334578434_The_M4_Competition_100000_time_series_and_61_forecasting_methods (дата звернення: 19.05.2025).

11. Wikipedia. Envelope system : електрон. ресурс. – URL: https://en.wikipedia.org/wiki/Envelope_system (дата звернення: 19.05.2025).

12. Wealthtender. Avoid using Elizabeth Warren's proposed personal budget plan : стаття, 2023. – URL: https://wealthtender.com/insights/money-management/avoid-using-elizabeth-warrens-proposed-personal-budget-plan/ (дата звернення: 19.05.2025).

13. Association of Chartered Certified Accountants. Comparing budgeting techniques : техн. стаття, 2023. – URL: https://www.accaglobal.com/gb/en/student/exam-support-resources/fundamentals-exams-study-resources/f5/technical-articles/comparing-budgeting-techniques.html (дата звернення: 19.05.2025).

14. Darden School of Business. Here's how to build a better personal budget : новина, 22.08.2024. – URL: https://news.darden.virginia.edu/2024/08/22/heres-how-to-build-a-better-personal-budget (дата звернення: 19.05.2025).

15. The Guardian. "Every penny has a purpose": the rise of zero-based budgeting : стаття, 20.04.2024. – URL: https://www.theguardian.com/money/2024/apr/20/every-penny-has-a-purpose-the-rise-of-zero-based-budgeting (дата звернення: 19.05.2025).

16. Yalantis. Personal finance app development : technology & marketplace : блог-пост, 2023. – URL: https://medium.com/yalantis-mobile/personal-finance-app-development-technology-marketplace-a113fd86b804 (дата звернення: 19.05.2025).

17. Revolut. Spending analytics : продуктова сторінка, 2024. – URL: https://www.revolut.com/blog/post/introducing-spending-analytics/ (дата звернення: 19.05.2025).

18. You Need A Budget (YNAB). Slaying the variable income dragon : блог-пост, 2024. – URL: https://www.youneedabudget.com/slaying-the-variable-income-dragon/ (дата звернення: 19.05.2025).

19. Trends Research & Advisory. Smart finance : how AI is shaping the future of budgeting and investing : аналіт. огляд, 2024. – URL: https://trendsresearch.org/insight/smart-finance-how-ai-is-shaping-the-future-of-budgeting-and-investing/?srsltid=AfmBOoqgOu1XiBsgRaHhJ_uuw6F5ROPcRlT0MH8BGhM6EmW68wkP9VAL (дата звернення: 19.05.2025).

20. Calibrate-Analytics. The crucial role of AI and machine learning in 2025 budget forecasting : блог-пост, 12.09.2024. – URL: https://calibrate-analytics.com/insights/2024/09/12/The-Crucial-Role-of-AI-and-Machine-Learning-in-2025-Budget-Forecasting/ (дата звернення: 19.05.2025).

21. Hyndman R. J. Short seasonal time-series forecasting with exponential smoothing : препринт, 2019. – URL: https://robjhyndman.com/papers/shortseasonal.pdf (дата звернення: 19.05.2025).

22. Boumezoued A. Seasonality strength in short series // Symmetry. – 2022. – Vol. 14, № 6. – URL: https://www.mdpi.com/2073-8994/14/6/1231 (дата звернення: 19.05.2025).

23. The evolution of CRISP-DM for data science : препринт, 2024. – URL: https://www.researchgate.net/publication/384999724_The_Evolution_of_CRISP-DM_for_Data_Science_Methods_Processes_and_Frameworks (дата звернення: 19.05.2025).

24. Yürek I. CRISP-DM : a comprehensive guide to the leading data-mining methodology : блог-стаття, 2024. – URL: https://medium.com/@ilyurek/crisp-dm-a-comprehensive-guide-to-the-leading-data-mining-methodology-396522b67477 (дата звернення: 19.05.2025).

25. Analytics India Mag. CRISP-DM : data-science project life-cycle : стаття, 2024. – URL: https://analyticsindiamag.com/ai-features/crisp-dm-data-science-project/ (дата звернення: 19.05.2025).

26. Modern Scientist. Time-series cross-validation for predictive modelling : блог-стаття, 2024. – URL: https://medium.com/the-modern-scientist/time-series-cross-validation-an-essential-technique-for-predictive-modeling-in-time-dependent-data-444693429eea (дата звернення: 19.05.2025).

27. NannyML. Population stability index (PSI) : how and why : блог-пост, 2024. – URL: https://www.nannyml.com/blog/population-stability-index-psi (дата звернення: 19.05.2025).

28. Makridakis S. The M3 competition : results, conclusions and implications : препринт, 2000. – URL: https://www.researchgate.net/publication/222514501_The_M3-Competition_Results_Conclusions_and_Implications (дата звернення: 19.05.2025).

29. Hyndman R. J. Forecasting: principles and practice : електрон. підручник, 2-ге вид., 2018. – URL: https://otexts.com/fpp2/ (дата звернення: 19.05.2025).

30. Investopedia. Box–Jenkins model : енциклопедична стаття, 2024. – URL: https://www.investopedia.com/terms/b/box-jenkins-model.asp (дата звернення: 19.05.2025).

31. Hyndman R. J. Complex seasonality : розділ електрон. книги, 2018. – URL: https://otexts.com/fpp2/complexseasonality.html (дата звернення: 19.05.2025).

32. Letham B., Taylor S. J. Forecasting at scale : Facebook Research PDF, 2017. – URL: https://lethalletham.com/ForecastingAtScale.pdf (дата звернення: 19.05.2025).

33. A data-driven ARIMA model selection alternative // Decision Support Systems, 2021. – URL: https://www.sciencedirect.com/science/article/pii/S0169207021001874 (дата звернення: 19.05.2025).

34. Interpretable deep learning for time-series forecasting // PLoS ONE, 2019. – URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889 (дата звернення: 19.05.2025).

35. Uber Engineering. What we learned from the M4 forecasting competition : блог-пост, 2018. – URL: https://www.uber.com/blog/m4-forecasting-competition/ (дата звернення: 19.05.2025).

36. Hyndman R. J. The Theta model : техн. звіт, 2011. – URL: https://robjhyndman.com/papers/Theta.pdf (дата звернення: 19.05.2025).

37. Assimakopoulos V., Nikolopoulos K. Theta : a decomposition approach to forecasting // European Journal of Operational Research, 2011. – URL: https://ideas.repec.org/a/eee/ejores/v214y2011i3p606-615.html (дата звернення: 19.05.2025).

38. Oleksios. Merchant Category Codes : JSON dataset : GitHub-репозиторій, 2024. – URL: https://github.com/Oleksios/Merchant-Category-Codes/blob/main/With%20groups/mcc-uk.json (дата звернення: 19.05.2025).

39. Deep learning for time-series forecasting : a review // Decision Support Systems, 2018. – URL: https://www.sciencedirect.com/science/article/pii/S0169207018301128 (дата звернення: 19.05.2025).

40. Hyndman R. J. Simple methods : розділ електрон. підручника FPP3, 2021. – URL: https://otexts.com/fpp3/simple-methods.html (дата звернення: 19.05.2025).

41. Hyndman R. J. Forecasting principles – GitHub resources : репозиторій, 2021. – URL: https://github.com/robjhyndman/forecasting-principles (дата звернення: 19.05.2025).

42. Hyndman R. J. M4 competition – resource page : електрон. ресурс, 2021. – URL: https://robjhyndman.com/publications/m4-competition (дата звернення: 19.05.2025).

43. LinkedIn. Why benchmarking forecasting models is essential : стаття, 2023. – URL: https://www.linkedin.com/pulse/why-benchmarking-forecasting-models-essential/ (дата звернення: 19.05.2025).

44. Penn State E-Learning. Seasonal ARIMA modelling (STAT 510, Lesson 4.1) : навч. матеріал, 2024. – URL: https://online.stat.psu.edu/stat510/lesson/4/4.1 (дата звернення: 19.05.2025).

45. GeeksforGeeks. Complete guide to SARIMAX in Python : навч. стаття, 2023. – URL: https://www.geeksforgeeks.org/complete-guide-to-sarimax-in-python/ (дата звернення: 19.05.2025).

46. Nixtla. StatsForecast – AutoETS documentation : техн. сторінка, 2024. – URL: https://nixtlaverse.nixtla.io/statsforecast/docs/models/autoets.html (дата звернення: 19.05.2025).

47. Hyndman R. J. Seasonal strength metric : розділ FPP2, 2018. – URL: https://otexts.com/fpp2/seasonal-strength.html (дата звернення: 19.05.2025).

48. Linking forecasting to inventory obsolescence // European Journal of Operational Research, 2007. – URL: https://www.sciencedirect.com/science/article/abs/pii/S0377221707011976 (дата звернення: 19.05.2025).

49. Syntetos A. A., Boylan J. E. A review of Croston's method for intermittent demand forecasting : препринт, 2010. – URL: https://www.researchgate.net/publication/254044245_A_Review_of_Croston%27s_method_for_intermittent_demand_forecasting (дата звернення: 19.05.2025).

50. Croston J. D. Forecasting and stock control for intermittent demand // Operational Research Quarterly, 1972. – URL: https://www.jstor.org/stable/3007885 (дата звернення: 19.05.2025).

51. Prestwich S. Forecasting for inventory management : презентація ISF 2015. – URL: https://forecasters.org/wp-content/uploads/gravity_forms/7-621289a708af3e7af65a7cd487aee6eb/2015/07/Prestwich_Steven_ISF2015.pdf (дата звернення: 19.05.2025).

52. Syntetos A. A., Boylan J. E. Intermittent demand – linking forecasting to inventory obsolescence // International Journal of Production Economics, 2005. – URL:

https://www.researchgate.net/publication/220288417_Intermittent_demand_Linking_forecasting_to_inventory_obsolescence (дата звернення: 19.05.2025).

53. Hyndman R. J. Prophet (FPP3) : розділ електрон. книги, 2021. – URL: https://otexts.com/fpp3/prophet.html (дата звернення: 19.05.2025).

54. Meta. Prophet – official documentation : електрон. ресурс, 2024. – URL: https://facebook.github.io/prophet/ (дата звернення: 19.05.2025).

55. Armstrong J. S., Collopy F. Error measures for general-purpose forecasting models // Management Science, 1992. – DOI: 10.1287/mnsc.38.6.829.

56. Makridakis S. Accuracy of extrapolation methods – results of a decade of research // International Journal of Forecasting, 1993. – DOI: 10.1016/0169-2070(93)90198-3.

57. Hyndman R. J., Koehler A. B. Another look at measures of forecast accuracy // International Journal of Forecasting, 2006. – DOI: 10.1198/073500104000000413.

58. Kaggle. M5 forecasting – accuracy competition : офіц. сторінка, 2020. – URL: https://www.kaggle.com/competitions/m5-forecasting-accuracy/overview (дата звернення: 19.05.2025).

59. Gneiting T., Raftery A. E. Strictly proper scoring rules, prediction and estimation // Journal of the American Statistical Association, 2007. – DOI: 10.1198/016214506000001437.

60. Bergmeir C., Benítez J. M. On the use of cross-validation for time-series predictor evaluation // Information Sciences, 2012. – DOI: 10.1016/j.ins.2012.02.028.

61. Hyndman R. J., Athanasopoulos G. Forecasting: principles and practice (FPP3) : електрон. підручник, 2021. – URL: https://otexts.com/fpp3 (дата звернення: 19.05.2025).

62. Syntetos A. A., Boylan J. E. The accuracy of intermittent demand estimates // International Journal of Forecasting, 2005. – DOI: 10.1016/j.ijpe.2004.12.009.

63. Shumway R. H., Stoffer D. S. Time series analysis and its applications : with R examples : 4-те вид., 2017. – URL: https://www.stat.pitt.edu/stoffer/tsa4 (дата звернення: 19.05.2025).

64. Brockwell P. J., Davis R. A. Introduction to time series and forecasting : 3-тє вид., 2015. – DOI: 10.1002/9781118675027.

65. Hyndman R. J. STL features : розділ FPP3, 2021. – URL: https://otexts.com/fpp3/stl-features.html (дата звернення: 19.05.2025).

66. Mermaid. Online editor – BPMN diagram (example link) : електрон. ресурс. – URL: https://mermaid.live/edit#pako:eNp1VNtq20AQ_ZVlIdBS31aSLxKJS9qmYLChWHowrUrZWmvZ... (дата звернення: 19.05.2025).

67. Microsoft. Azure Logic Apps – overview : документація, 2025. – URL: https://learn.microsoft.com/en-us/azure/logic-apps/logic-apps-overview (дата звернення: 19.05.2025).

68. Microsoft. Azure Data Factory – introduction : документація, 2025. – URL: https://learn.microsoft.com/en-us/azure/data-factory/introduction (дата звернення: 19.05.2025).

69. Cloudprice.net. Azure VM Standard_D2s_v3 pricing calculator : електрон. ресурс. – URL: https://cloudprice.net/vm/Standard_D2s_v3 (дата звернення: 19.05.2025).

70. Microsoft Azure. Windows virtual machines – pricing : веб-сторінка, 2025. – URL: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/ (дата звернення: 19.05.2025).

71. Microsoft Power BI. Important update to Power BI pricing : блог-пост, 2024. – URL: https://powerbi.microsoft.com/en-us/blog/important-update-to-microsoft-power-bi-pricing/ (дата звернення: 19.05.2025).

72. American Friends of Hebrew University. How AI can help consumers curb overdraft costs : прес-реліз, 16.05.2025. – URL: https://www.afhu.org/2025/05/16/new-study-shows-how-ai-can-help-consumers-

curb-overdraft-costs-according-to-hebrew-university-researchers/ (дата звернення: 19.05.2025).

73. FinHealth Network. Overdraft & NSF fees – bigger burden than previously estimated : звіт, 2023. – URL: https://finhealthnetwork.org/research/overdraft-nsf-fees-bigger-burden-than-previously-estimated/ (дата звернення: 19.05.2025).

74. Microsoft. Azure OpenAI Service – overview : документація, 2025. – URL: https://learn.microsoft.com/en-us/azure/ai-services/openai/overview (дата звернення: 19.05.2025).

75. Microsoft. Azure OpenAI – data privacy : документація, 2025. – URL: https://learn.microsoft.com/en-us/legal/cognitive-services/openai/data-privacy (дата звернення: 19.05.2025).

76. Erasmus University Rotterdam. Master thesis N°304 : електрон. ресурс, 2024. – URL: https://thesis.eur.nl/pub/66680/-304 (дата звернення: 19.05.2025).

77. PeerJ. Interpretable deep learning for time-series forecasting : стаття, 2019. – URL: https://peerj.com/articles/7056/ (дата звернення: 19.05.2025).

ДОДАТКИ

ДОДАТОК А. Програмний код реалізації

Файл 01 – data preparation.ipynb

1) Завантаження даних транзакцій

Monobank: транзакції експортовані з додатку (CSV)

```
from pathlib import Path
import pandas as pd

data_dir = Path("transactions_data")

# Пошук усіх файлів *.csv
csv_files = list(data_dir.glob("report*.csv"))

# Прочитати їх у список DataFrame
dfs = []
for fp in csv_files:
    df = pd.read_csv(fp)
    # за потреби можна додати стовпець із іменем файлу:
    # df["source_file"] = fp.name
    dfs.append(df)

# Об’єднання всього в один DataFrame
monobank_source = pd.concat(dfs, ignore_index=True)

# Перевірка результату
print(monobank_source.shape)
monobank_source.head()
```

ПриватБанк: транзакції експортовані з додатку (CSV)

```
from pathlib import Path
import pandas as pd

data_dir = Path("transactions_data")

# Пошук усіх файлів *.csv
csv_files = list(data_dir.glob("privat*.csv"))

# Прочитайте їх у список DataFrame
dfs = []
for fp in csv_files:
    df = pd.read_csv(fp)
    # за потреби можна додати стовпець із іменем файлу:
    # df["source_file"] = fp.name
    dfs.append(df)

# Об’єднайте все в один DataFrame
pryvatbank_source = pd.concat(dfs, ignore_index=True)

# Перевірте результат
print(pryvatbank_source.shape)
pryvatbank_source.head()
```

2) Завантаження додаткових даних

Завантаження кодів MCC з репозиторію

```
import pandas as pd

# 1) Load the JSON straight into a DataFrame
url = "https://raw.githubusercontent.com/Oleksios/Merchant-Category-
Codes/refs/heads/main/With%20groups/mcc-uk.json"
df_mcc = pd.read_json(url)          # df_mcc.columns -> ['mcc', 'group',
'shortDescription', 'fullDescription']

# 2) Split the `group` dict into its own columns …
group_cols = df_mcc['group'].apply(pd.Series)           # columns:
['type', 'description']

# 3) … and combine everything
df_mcc_flat = pd.concat(
    [df_mcc.drop(columns='group'),                      # keep the
original scalar columns
     group_cols.rename(columns={                    # rename for clarity
(optional)
         'type':        'groupAbbr',
         'description': 'groupDescription'
     })],
    axis=1
)

df_mcc_flat.to_csv(ADDITIONAL_DATA_DIR+"/mcc_codes.csv")
df_mcc_flat.head()
```

3) Підготовка даних

Підготовка транзакцій Monobank

```
monobank_transactions = monobank_source.rename(columns={
    "Дата і час операції": "datetime",
    "Деталі операції": "description",
    "Сума в валюті картки (UAH)": "amount_in_uah",
    "Сума в валюті операції": "amount_in_currency",
    "Валюта": "currency",
    "MCC": "mcc"})
monobank_transactions['datetime'] = pd.to_datetime(
    monobank_transactions['datetime'],
    dayfirst=True,
    format="%d.%m.%Y %H:%M:%S"
)
# Об'єднання по стовпцю 'mcc'
mono_mcc_transactions = pd.merge(monobank_transactions, df_mcc_flat,
how='left', left_on='mcc', right_on='mcc')
mono_mcc_transactions['type'] =
mono_mcc_transactions['amount_in_uah'].apply(lambda x: 'income' if x > 0
else 'spend')
mono_mcc_transactions['bank'] = 'Monobank'
mono_mcc_transactions.rename(columns={
    "shortDescription": "category",
    }, inplace=True)
```

```
# Виведення об'єднаних даних
mono_mcc_transactions = mono_mcc_transactions[["datetime", "mcc",
"description", "amount_in_uah", "amount_in_currency", "currency",
"category", "type", "bank"]]
mono_mcc_transactions
```

Підготовка транзакцій Monobank

```
import pandas as pd

pryvatbank_transactions = pryvatbank_source.copy()

pryvatbank_transactions["datetime"] = pd.to_datetime(
    pryvatbank_transactions["date"] + " " +
pryvatbank_transactions["time"],
    dayfirst=True,
    format="%d.%m.%Y %H:%M"
)

s = pryvatbank_transactions["amount_uah"]

pryvatbank_transactions["amount_uah"] = (
    pryvatbank_transactions["amount_uah"]
    .str.replace(" ", "", regex=False)   # remove thousands-sep
    .str.replace(",", ".", regex=False)  # make a Python-style decimal
point
    .astype(float)                       # convert to float
)
pryvatbank_transactions["amount_orig"] = (
    pryvatbank_transactions["amount_orig"]
    .str.replace(" ", "", regex=False)   # remove thousands-sep
    .str.replace(",", ".", regex=False)  # make a Python-style decimal
point
    .astype(float)                       # convert to float
)

pryvatbank_transactions['category'] =
pryvatbank_transactions['description'].str.extract(r'^([^.,#:]+)\s*:',
expand=False).str.strip()

pryvatbank_transactions.rename(columns={
    "amount_uah": "amount_in_uah",
    "amount_orig": "amount_in_currency",
    "currency_orig": "currency"}, inplace=True)
pryvatbank_transactions['bank'] = 'ПриватБанк'
pryvatbank_transactions['type'] =
pryvatbank_transactions['amount_in_uah'].apply(lambda x: 'income' if x >
0 else 'spend')

pryvatbank_transactions = pryvatbank_transactions[["datetime",
"description", "amount_in_uah", "amount_in_currency", "currency", "bank",
"type", "category"]]
pryvatbank_transactions
```

4) Об'єднання транзакцій Monobank та ПриватБанк у єдиний датафрейм

```
# Об'єднання DataFrame по рядках
transactions_df = pd.concat([pryvatbank_transactions,
mono_mcc_transactions],
ignore_index=True).sort_values(by='datetime').reset_index(drop=True)
transactions_df.to_csv("transactions_union\\transactions.csv")

# Перевірка результату
transactions_df.sample(10)
```

Прибираємо транзакції, які були скасовані

```
import pandas as pd

# Встановлюємо допустиме відхилення (наприклад, 5 гривень або 1%)
tolerance = 0.05  # або можна встановити відсоткове відхилення,
наприклад, 0.01 для 1%

# Фільтруємо транзакції, де type == 'income' і description містить
'Скасування. '
income_cancellations = transactions_df[
    (transactions_df['type'] == 'income') &
    (transactions_df['description'].str.contains('Скасування.'))
]

# Створюємо список описів без "Скасування. "
income_cancellations['description_clean'] =
income_cancellations['description'].str.replace('Скасування. ', '',
regex=False)

# Знаходимо відповідні транзакції з type == 'spend', де description і
сума відповідають умовам з урахуванням відхилення
matching_spends = []

for idx, spend_row in transactions_df[transactions_df['type'] ==
'spend'].iterrows():
    spend_description = spend_row['description']
    spend_amount = spend_row['amount_in_uah']
    spend_datetime = spend_row['datetime']

    # Фільтруємо доходи з тим самим описом і відповідною сумою
    matched_income = income_cancellations[
        (income_cancellations['description_clean'] == spend_description)
&
        ((income_cancellations['amount_in_uah'] == spend_amount * -1) |
((income_cancellations['amount_in_uah'] > 1500) & (1 -
abs(income_cancellations['amount_in_uah'] / spend_amount) <= tolerance)))
&
        (income_cancellations['datetime'] >= spend_datetime) &
        (income_cancellations['datetime'] <= spend_datetime +
pd.Timedelta(days=7))
    ]

    if not matched_income.empty:
        matching_spends.append(spend_row)

# Перетворюємо список знайдених транзакцій назад у DataFrame
matching_spends_df = pd.DataFrame(matching_spends)
```

```
# Перевірка результату
matching_spends_df.head(10)

transactions_without_cancelling =
transactions_df[~transactions_df.index.isin(matching_spends_df.index)]
```

Прибираємо транзакції-доходи

```
import re

spends_only =
transactions_without_cancelling[transactions_without_cancelling["type"]
== "spend"]

spends_only = spends_only.loc[
    ~(spends_only['description']
      .str.contains('ПЕРЕНОС % ЗА ПОЛОЖ', na=False) |
spends_only['description']
      .str.contains('«Захист на кожен день»', na=False))
].reset_index(drop=True)

spends_only
```

Групуємо категорії у декілька загальних груп (файл custom categories.xlsx)

```
import pandas as pd

# Крок 1: Прочитати Excel-файл
categories_df = pd.read_excel('supplementary_data\custom
categories.xlsx')  # Вкажіть шлях до вашого файлу Excel

# Крок 2: Прочитати DataFrame з транзакціями (з другого скріншоту)
# transactions_without_cancelling - DataFrame з транзакціями, який ви вже
маєте

# Крок 3: Об'єднати DataFrame, щоб додати новий стовпець на основі
категорій
# Створимо словник для мапінгу категорій на їх спільні назви
category_mapping = categories_df.set_index('category')['common category
(custom)'].to_dict()

# Додаємо новий стовпець 'common_category' до
transactions_without_cancelling на основі мапінгу категорій
transactions_with_categories = spends_only.copy()
transactions_with_categories['common_category'] =
transactions_with_categories['category'].map(category_mapping)
transactions_with_categories[~transactions_with_categories['common_catego
ry'].isna()].sample(15)
transactions_with_categories[transactions_with_categories['category'].isn
a()].sample(20)
```

Визначаємо категорії за описом транзакції (description_category.xlsx)

```
transactions__categ_from_desc = transactions_with_categories.copy()

# Крок 1: Зчитати файл description_category.xlsx
description_category_df =
pd.read_excel('supplementary_data\description_category.xlsx')
```

```
# Крок 2: Ітеруємося по рядках цього файлу
for index, row in description_category_df.iterrows():
    keyword = row['description contains']
    new_category = row['category']
    print(f"{keyword} -> {new_category}")

    # Крок 3: Оновлюємо колонку common_category у основному DataFrame

transactions__categ_from_desc.loc[transactions__categ_from_desc['descript
ion'].str.contains(keyword, na=False), 'common_category'] = new_category
```

3) Екзогенні змінні

Отримуємо та готуємо датасет з індексом споживчих цін

```
import requests
import pandas as pd

# 1. Завантажуємо сторінку з ІСЦ
url = "https://ukrstat.gov.ua/imf/arhiv/isc_u.htm"
resp = requests.get(url)
resp.encoding = 'cp1251'

# 2. Парсимо всі таблиці, беремо першу
cpi_wide = pd.read_html(resp.text)[0]

# 3. Переіменовуємо перший стовпець у 'year'
cpi_wide = cpi_wide.rename(columns={cpi_wide.columns[0]: 'year'})
cpi_wide = cpi_wide[cpi_wide["year"] > "2020"]

for col in cpi_wide.columns:
    cpi_wide[col] = pd.to_numeric(cpi_wide[col],
errors='coerce').astype('Int64')

cpi_wide['year'] = cpi_wide['year'] // 10

# 4. «Розплавляємо» стовпці місяців у довгий формат
cpi_long = cpi_wide.melt(
    id_vars=['year'],
    var_name='month_number',
    value_name='cpi_wide_index'
)

cpi_long.to_csv("supplementary_data\\cpi.csv")

cpi_long.head()
```

Додаємо екзогенні змінні до набору даних

```
import pandas as pd
import requests
import holidays

# === 1. Транзакції
=====================================================
prepared_transactions = pd.read_csv(
    r'prepared_transactions\transactions.csv',
    parse_dates=['datetime']
)
```

```
# period-колонка для CPI-merge
prepared_transactions['month_period'] = (
    prepared_transactions['datetime'].dt.to_period('M')
)
# окремі числові колонки
prepared_transactions['year'] = prepared_transactions['datetime'].dt.year
prepared_transactions['month_num'] =
prepared_transactions['datetime'].dt.month
prepared_transactions['day_of_month'] =
prepared_transactions['datetime'].dt.day

# === 2. Курс USD
============================================================
rate_records = []
for p in prepared_transactions['month_period'].unique():
    last_day = p.to_timestamp('M').strftime('%Y%m%d')
    url =
f'https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?date={last
_day}&json'
    usd_rate = next(
        (x['rate'] for x in requests.get(url).json() if x.get('cc') ==
'USD'),
        None
    )
    rate_records.append({'month_period': p, 'usd_rate': usd_rate})

df_fx = pd.DataFrame(rate_records)
df_fx['month_code'] = df_fx['month_period'].astype(str)   # YYYY-MM
(object)

# === 3. CPI
================================================================
df_cpi = pd.read_csv(r'supplementary_data\cpi.csv')       # year,
month_number …
df_cpi['month_period'] = pd.PeriodIndex(
    year=df_cpi['year'], month=df_cpi['month_number'], freq='M'
)
df_cpi_monthly = (
    df_cpi
    .rename(columns={'cpi_wide_index': 'cpi_index'})
    [['month_period', 'cpi_index']]
)

# === 4. Державні свята
=========================================================
ua_holidays =
holidays.Ukraine(years=prepared_transactions['year'].unique())
prepared_transactions['is_public_holiday'] = (
    prepared_transactions['datetime'].isin(ua_holidays).astype(int)
)

# === 5. Особисті свята
=========================================================
personal = (
    pd.read_excel(r'supplementary_data\holidays.xlsx')   # holiday,
month, day
      .rename(columns={'month': 'month_num', 'day': 'day_of_month'})
)
```

```
prepared_transactions = (
    prepared_transactions
      .merge(personal, on=['month_num', 'day_of_month'], how='left')
      .assign(is_personal_holiday=lambda d:
d['holiday'].notna().astype(int))
      .drop(columns=['holiday'])
)

# === 6. month_code для USD-merge
=========================================
prepared_transactions['month_code'] = (
    prepared_transactions['year'].astype(str)
    + '-'
    + prepared_transactions['month_num'].astype(str).str.zfill(2)
)

# === 7. Фінальне об’єднання
================================================
df_final = (
    prepared_transactions
      .merge(df_fx[['month_code', 'usd_rate']], on='month_code',
how='left')
      .merge(df_cpi_monthly, on='month_period', how='left')
)

df_final.to_csv(
    r'prepared_transactions\transactions_with_exog.csv',
    index=False,
    encoding='utf-8'
)

df_final
```

Файл 02 – data_analysis.ipynb

```
# Cell 1 ──────────────────────────────────────────────────────────────
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# завантаження даних
df = pd.read_csv(r"prepared_transactions\transactions.csv",
parse_dates=["datetime"])

# базова підготовка
#df = df[df["datetime"] >= "2021-07-01"]                    # 2-га
половина 2020-го
df = df[~df["common_category"].str.lower().eq("заощадження")]  #
виключаємо
df["amount_uah"] = df["amount_in_uah"].abs()
df["month"] = df["datetime"].dt.to_period("M")

# Cell 2 ─── Бар-чарт суми за місяцями
monthly_spend = df.groupby("month")["amount_uah"].sum()

plt.figure()
monthly_spend.plot(kind="bar")
plt.title("Місячні витрати (UAH)")
```

```
plt.xlabel("Місяць")
plt.ylabel("Сума, ₴")
plt.xticks(rotation=90, ha="center")
# plt.tight_layout()
plt.figure(figsize=(20, 5))
plt.show()

# Cell 3 ————————————————————————————————————————————————————————
plt.figure(figsize=(10, 5))
monthly_spend.plot()
plt.title("Місячні витрати")
plt.xlabel("Місяць")
plt.ylabel("Сума, ₴")
plt.tight_layout()
plt.show()

# Cell 4 ——— Стек-бар (кількість транзакцій)
bank_month_cnt = (
    df.groupby(["month", "bank"]).size().unstack(fill_value=0)
)

plt.figure()
bank_month_cnt.plot(kind="bar", stacked=True)
plt.title("Кількість транзакцій за банками (помесячно)")
plt.xlabel("Місяць")
plt.ylabel("Кількість")
plt.xticks(rotation=90, ha="center")
plt.tight_layout()
plt.show()

# Cell 5 – Heatmap
heat = (
    df.pivot_table(columns="month",
                   index="common_category",
                   values="amount_uah",
                   aggfunc="sum")
)

plt.figure(figsize=(12, 6))                    # 1) більша висота
ax = sns.heatmap(
    heat,
    cmap="jet",
    linewidths=0.5,
    linecolor="lightgray",
    square=True,
    cbar_kws={"shrink": 0.5}                    # за бажанням зменшити
colorbar
)

# 2) менший шрифт та орієнтація 0° для підписів категорій
ax.set_yticklabels(
    ax.get_yticklabels(),
    rotation=0,
    fontsize=8
)

# 3) додатковий відступ зліва, щоб підписи не обрізалися
plt.subplots_adjust(left=0.25, bottom=0.1)
```

```
ax.set_title("Теплова карта: витрати по місяцях та категоріях")
ax.set_xlabel("Місяць")
ax.set_ylabel("Категорія")
plt.tight_layout()
plt.show()

# Cell 6 ──────────────────────────────────────────────────────────────
cat_totals = (
    df.groupby("common_category")["amount_uah"]
      .sum()
      .sort_values(ascending=False)
)

plt.figure()
cat_totals.plot(kind="bar")
plt.title("Сумарні витрати за категоріями (2020-2025)")
plt.ylabel("Сума, ₴")
plt.xticks(rotation=45, ha="center")
plt.tight_layout()
plt.show()

# Cell 7 ──────────────────────────────────────────────────────────────
bank_share = (
    df.groupby("bank")["amount_uah"]
      .sum()
      .sort_values(ascending=False)
)

plt.figure()
bank_share.plot(kind="pie", autopct="%1.1f%%")
plt.title("Частка витрат за банками")
plt.ylabel("")   # ховаємо зайву вісь
plt.tight_layout()
plt.show()

# Cell 8 ─── Бар-чарт із «роздільниками» між роками
plt.figure()
ax = monthly_spend.plot(kind="bar")

prev_year = monthly_spend.index[0].year
for i, prd in enumerate(monthly_spend.index):
    if prd.year != prev_year:
        ax.axvline(i - 0.5, linestyle="--", linewidth=1)  # пунктир-
роздільник
        prev_year = prd.year

plt.title("Місячні витрати (розбивка за роками)")
plt.xlabel("Місяць")
plt.ylabel("Сума, ₴")
plt.xticks(rotation=45, ha="center")
plt.tight_layout()
plt.show()

# Celll  8 Місячні витрати за категоріями ──────────────────────────
# 1. Зводимо суму витрат за кожен місяць та категорію
pivot_all = (
    df.pivot_table(index="month",
```

```
                    columns="common_category",
                    values="amount_uah",
                    aggfunc="sum")
)

# 2. Впорядковуємо стовпці за алфавітом
pivot_all = pivot_all[sorted(pivot_all.columns, key=str.lower)]

# 3. Будуємо велику фігуру
plt.figure(figsize=(25, 8))                 # ← за потреби змінюйте
розмір
ax = pivot_all.plot(kind="bar", stacked=True, width=0.9)

# 4. Оформлення
plt.title("Місячні витрати за категоріями", fontsize=14)
plt.xlabel("Місяць")
plt.ylabel("Сума, ₴")
plt.xticks(rotation=45, ha="right")

# Менший шрифт легенди та розміщення поза областю графіка
ax.legend(fontsize=8, bbox_to_anchor=(1.02, 1), loc="upper left",
borderaxespad=0)

plt.tight_layout()
plt.show()
```

ДОДАТОК Б. Алгоритм диспетчера, що обирає підходящу модель

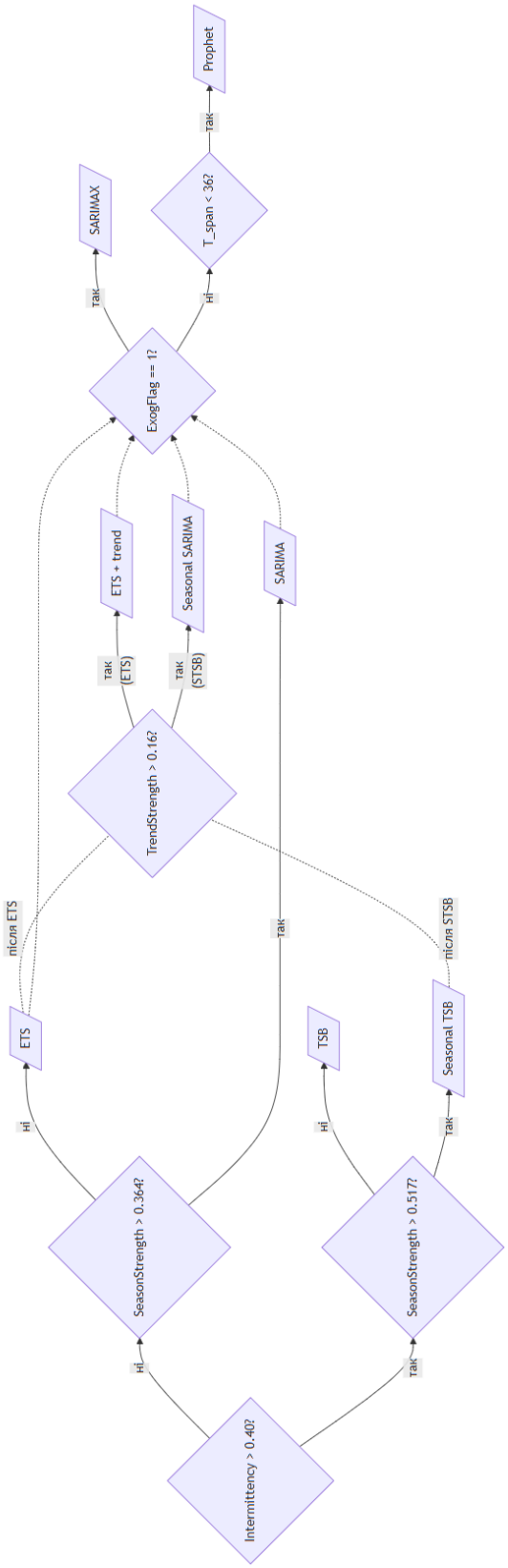

Рисунок Б.1. Діаграма алгоритму диспетчера