

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)

Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня магістра

Студента _____ Андрєєва Іллі Володимировича _____

академічної групи _____ 124м – 22 – 1 _____

спеціальності _____ 124 Системний аналіз _____

на тему: «Дослідження методів аналізу та прогнозування часових рядів роздрібних продажів»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг овою	Інституційною	
кваліфікаційн ої роботи	<i>к.т.н., доц. Желдак Т.А.</i>			
розділів:				
Інформаційно- аналітичний	<i>к.т.н., доц. Желдак Т.А.</i>			
Спеціальн ий розділ	<i>к.т.н., доц. Желдак Т.А.</i>			
Рецензент				
Нормоконтролер	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
Системного аналізу та управління

(повна назва)

_____ к.т.н., доц. Желдак Т.А.
(підпис) (прізвище, ініціали)

« _____ » _____ 20 ____ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра

студенту Андрєєву І. В. академічної групи 124м-22-1

спеціальності: 124 Системний аналіз

на тему «Дослідження методів аналізу та прогнозування часових рядів
роздрібних продажів»

затверджену наказом ректора НТУ «Дніпровська політехніка»
від 09.10.2023 р. №1227-с

Розділ	Зміст	Терміни виконання
1. Інформаційно-аналітичний розділ	<i>Проаналізувати структуру об'єкта дослідження. Визначити предметну область дослідження та проблему, що розв'язується. Обґрунтувати методи виконання поставлених завдань</i>	04.09.2023 – 18.10.2023
2. Спеціальний розділ	<i>Розв'язати поставлені задачі: проаналізувати та спрогнозувати прибуток магазину, який займається роздрібними продажами</i>	18.10.2023 – 30.11.2023

Завдання видано _____ доц. Желдак Т.А.
(підпис) (прізвище, ініціали)

Дата видачі: 04.09.2023 р.

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Андрєєв І. В.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 58 с., 15 рис., 1 табл., 3 додатки, 17 джерел.

Об'єктом дослідження в роботі є процес прогнозування прибутку за умов нестійкого попиту з прикладу магазину одягу.

Предметом дослідження є сукупність теоретико-методичних та науково-практичних аспектів аналізу та прогнозування часових рядів роздрібних продажів.

Мета даної кваліфікаційної роботи – це розглянути теоретичні основи прогнозування часових рядів, а також вивчення економіко-математичного апарату аналізу та прогнозування прибутку магазину.

Методи дослідження: методи об'єктно – орієнтованого аналізу та прогнозування часових рядів, принципи об'єктно – орієнтованого програмування на мові Python.

В *інформаційно–аналітичному розділі* наведено аналіз об'єкту дослідження та ключових проблем на ньому. Поставлені задачі дослідження та обрано концепції їх розв'язання.

У *спеціальному розділі* сформовано алгоритм програми для аналізу поставленої задачі, написано програмний код для розв'язання існуючої проблеми.

Практична цінність отриманих результатів полягає в тому, що запропонована розроблена система дає можливість своєчасно виявити надлишок або нестачу прибутку.

Ключові слова: АНАЛІЗ, ПРОГНОЗУВАННЯ, МОДЕЛЬ, ПРИБУТОК, ПРОДАЖ, РОЗДРІБНА ТОРГІВЛЯ, МГУА, ШТУЧНИЙ ІНТЕЛЕКТ.

ABSTRACT

Explanatory note: 58 pages, 15 figures, 1 tables, 3 appendices, 17 sources.

The object of the study is the process of forecasting profits in conditions of volatile demand, for example, a clothing store

The subject of research is a set of theoretical and methodological and scientific and practical aspects of analysis and forecasting of time series of retail sales

The purpose of this qualification work is to consider the theoretical foundations of time series forecasting, as well as the study of economic and mathematical apparatus of analysis and forecasting of store profits.

Research methods: methods of object-oriented analysis and forecasting of time series, principles of object-oriented programming in Python.

The information-analytical section provides an analysis of the research object and key problems on it. The tasks of the research are set and the concepts of their solution are chosen.

In a special section, the algorithm of the program for the analysis of the task is formed, the program code for solving the existing problem is written.

The practical value of the obtained results is that the proposed developed system makes it possible to identify excess or lack of profit in a timely manner.

Keywords: ANALYSIS, FORECASTING, MODEL, PROFIT, SALES, RETAIL, GMDH, ARTIFICIAL INTELLIGENCE.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ.....	8
1.1 Часовий ряд у роздрібних продажах. Що ми хочемо проаналізувати та передбачити?.....	8
1.2 Основні кроки аналізу часових рядів.....	10
1.3 Попередня обробка та очищення даних часових рядів.....	13
1.3.1 Очищення даних.....	13
1.3.2 Перетворення даних.....	13
1.3.3 Інтеграція даних.....	13
1.3.4 Скорочення даних.....	13
1.4 Функціональна інженерія.....	14
1.4.1 Особливості дати і часу.....	15
1.4.2 Перетворення категоріальних ознак.....	15
1.4.3 Перетворення тексту.....	15
1.4.4 Для предметної області.....	17
1.5 Перевірка моделі та аналіз результатів.....	18
1.5.1 Показники точності прогнозу часових рядів.....	18
1.5.2 Метод перехресної перевірки для часових рядів.....	19
1.6 Метод групового урахування аргументів.....	20
1.7 Висновки до розділу 1.....	25
РОЗДІЛ 2 СПЕЦІАЛЬНИЙ.....	26
2.1 Постановка задачі.....	26
2.1.1 Завантаження даних та бібліотек.....	26
2.1.2 Виявлення та коригування аномалій.....	31
2.1.3 Декомпозиція на основі тренду та сезонності.....	33
2.1.4 Тренувальні та тестові набори.....	34
2.2 Модель Хольта – Вінтерса.....	35
2.3 Модель SARIMA.....	38
2.4 Модель множинної лінійної регресії.....	42

	6
2.5 Метод групового урахування аргументів.....	47
2.6 Висновки до розділу 2.....	50
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
Додаток А. Відомість матеріалів кваліфікаційної роботи.....	55
Додаток Б. Відгук керівника.....	56
Додаток В. Прибуток магазину за 5 років.....	57

ВСТУП

Збільшення переліку товарів для продажу спонукало багато комерційних організацій шукати нові підходи до прогнозування продажів. Можна було передбачити продаж десятків товарів за допомогою детального аналізу, але цього не можна було зробити за сотні, особливо тисячі серій. У той же час, для будь-якої компанії, що шукає місце на ринку, необхідно зробити прогнози продажів для всіх, або хоча б для найпоширеніших видів товарів, інакше вони б зазнали значних фінансових втрат.

Головною особливістю цих завдань є наявність бази даних продажів для широкого асортименту товарів (від сотень до десятків тисяч). Це означає, що існує багато періодів реалізації, які вимагають регулярного продовження бюджетного планування та управлінських рішень.

Актуальність даної теми полягає в тому, що при нинішньому нестабільному стані економіки (найбільше спричиненою економічною кризою) багато фірм, що займаються торговельною діяльністю, зазнають збитків через занижену або, що зустрічається частіше, завищену кількість продукції на складі. Тому виникла потреба у глибокому вивченні методів прогнозування, їх ефективності та шляхів реалізації.

РОЗДІЛ 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ

1.1 Часовий ряд у роздрібних продажах. Що ми хочемо проаналізувати та передбачити?

Серед основних напрямів науки про дані виділили 10 пунктів, пов'язаних саме з роздрібними продажами. Як правило, компанія, яка займається такими завданнями, працює над рекомендаційними системами (розуміти, що людина купуватиме і що їй можна порекомендувати), працює у рамках персоналізованого маркетингу, оптимізації цінової політики товарів. Дуже активно аналізується кількість запасів товарів тощо.

- Рекомендаційна система
- Персоналізований маркетинг
- Оптимізація цін
- Управління запасами
- Доповнена реалізація реальності
- Місце розташування нових магазинів
- Аналіз споживчих настроїв
- Використання соціальних медіа
- Виявлення шахрайства
- Мерчендайзинг

Де ми можемо отримати дані для аналізу?

Звичайно, деякі дані збираються і накопичуються на деяких відкритих ресурсах. Однак багато компаній, що займаються комерційною аналітикою, можуть збирати дані з сайту за допомогою спеціальних систем і формувати їх як DataSet'и, де ми можемо отримати повний опис продукту (малюнок продукту, ціна, характеристики, кількість оглядів тощо).

Дані роздрібних продажів можуть бути дуже різними.

Змінні можуть бути представлені в дуже різних шкалах. Це можуть бути метричні ознаки. Можна використовувати багато параметричної статистики і, звичайно, класичні методи обробки даних. Існує безліч категоріальних змінних (коли був проданий товар, колір товару, матеріал, з якого виготовляється товар тощо). Звичайно, буде багато текстового опису, що багато в чому впливає на продаж товарів.

Дані можуть бути структуровані і неструктуровані. Вони можуть бути на різних рівнях: місяці, тижні, дні, години...

- Чисельна змінна
- Категоріальні змінні
- Текст
- Структурований або неструктурований
- Рівні: час, день, тиждень або місяць
- Панельні або ієрархічні дані
- Узагальнені або детальні дані

Коли ми аналізуємо роздрібні продажі, найголовніше - зрозуміти мету до якої ми прагнемо. “Гарне прогнозування допомагає гарантувати, що роздрібні торговці можуть доставити правильний продукт в потрібний час і в потрібному місці, підтримувати достатній рівень запасів, уникаючи при цьому дефіциту, зменшити ймовірність старіння запасів і поліпшити управління ціною і просуванням”, - Irad Ben Gal.

Часовий ряд — це деяка послідовність значень, що відповідають певному моменту в часі. Будь-які дані, які ми збираємо у часі, як-от, зміни температури, попит на продукти, мінеральне виробництво, можуть бути представлені у вигляді часового ряду, якщо ми точно знаємо, коли в нас з’явилося те або інше значення.

1.2 Основні кроки аналізу часових рядів

При аналізі часових рядів слід дотримуватися певного алгоритму. Звичайно, на першому етапі завжди буде аналіз вимог клієнтів. Саме на основі цих вимог ми повинні вирішити, як збирати дані. Дуже часто завдання стають так, що клієнт сам представляє частину інформації і дає дані про продаж свого товару. Але іноді цих даних досить мало, тому необхідно проаналізувати дані, але зрозуміти, звідки ми можемо взяти додаткові, проте щоб не було зайвої надмірності (не брати того, що не корисно в аналізі)/

1. Аналіз вимог
2. Збір даних
3. Аналіз функцій та даних
4. Попередня обробка даних
5. Інженерія функцій
6. Моделювання
7. Традиційні методи 7.1. Методи ML
8. Перевірка моделі
9. Розгортання моделі

Класифікація рядів: стаціонарна або нестаціонарна; лінійна або нелінійна залежності; сезонність або не сезонність; одномірний або багатомірний.

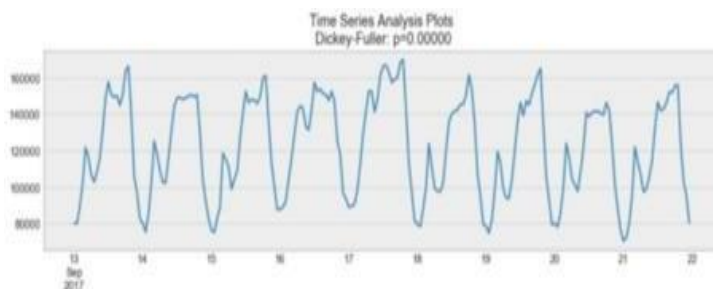


Рис. 1.1. Стаціонарна залежність

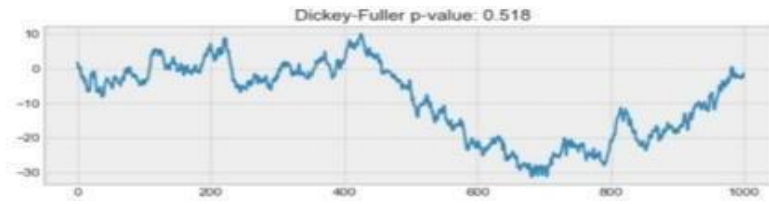


Рис. 1.2. Нестационарна залежність

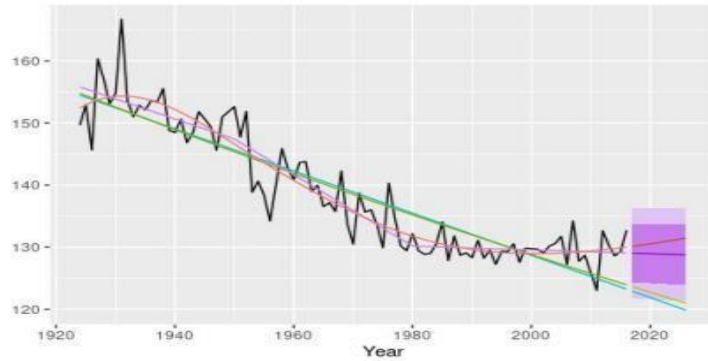


Рис. 1.3. Лінійна та нелінійна залежності

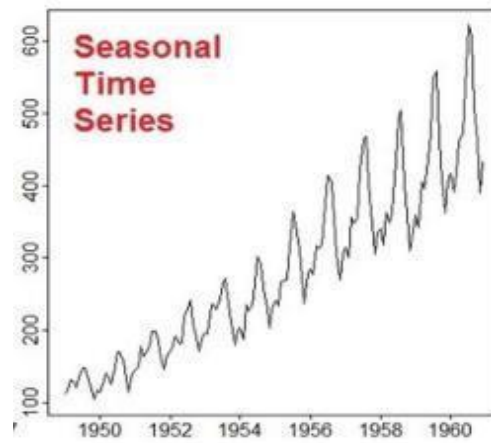


Рис. 1.4. Приклад сезонності

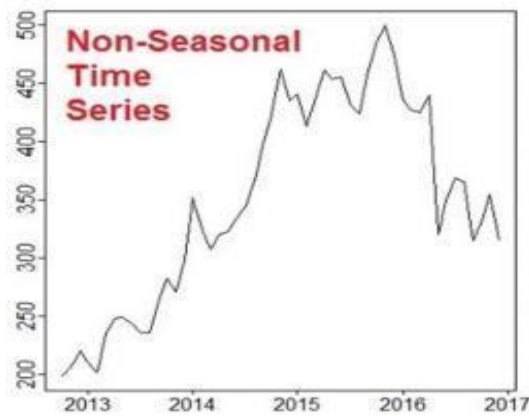



Рис. 1.5. Приклад не сезонності



Time	Temperature
5:00 am	59 °F
6:00 am	59 °F
7:00 am	58 °F
8:00 am	58 °F
9:00 am	60 °F
10:00 am	62 °F
11:00 am	64 °F
12:00 pm	66 °F
1:00 pm	67 °F
2:00 pm	69 °F
3:00 pm	71 °F
4:00 pm	71 °F
5:00 pm	71 °F
6:00 pm	69 °F

Рис. 1.6. Одномірний ряд



Time	Temperature	cloud cover	dew point	humidity	wind
5:00 am	59 °F	97%	51 °F	74%	8 mph SSE
6:00 am	59 °F	89%	51 °F	75%	8 mph SSE
7:00 am	58 °F	79%	51 °F	76%	7 mph SSE
8:00 am	58 °F	74%	51 °F	77%	7 mph S
9:00 am	60 °F	74%	51 °F	74%	7 mph S
10:00 am	62 °F	74%	52 °F	70%	8 mph S
11:00 am	64 °F	76%	52 °F	65%	8 mph SSW
12:00 pm	66 °F	80%	52 °F	60%	8 mph SSW
1:00 pm	67 °F	78%	52 °F	58%	10 mph SW
2:00 pm	69 °F	71%	52 °F	54%	10 mph SW
3:00 pm	71 °F	75%	52 °F	52%	11 mph SW
4:00 pm	71 °F	78%	52 °F	52%	11 mph SW
5:00 pm	71 °F	78%	52 °F	52%	12 mph SW
6:00 pm	69 °F	78%	52 °F	54%	11 mph SW

Рис. 1.7. Багатомірний ряд

1.3 Попередня обробка та очищення даних часових рядів

1.3.1 Очищення даних

- Оцінка відсутнього значення
- Аналіз дублікатів
- Ідентифікація викидів у часовому ряді
- Календар (населення, інфляція та ін.) регулювання

1.3.2 Перетворення даних

- Трансформація потужності
- Дельта-перетворення (згладжування)
- Дискретність
- Нормалізація
- Агрегація

1.3.3 Інтеграція даних

- Інтегрування схем
- Проблема ідентифікації об'єктів
- Виявлення і роздільна здатність концепцій значення даних

1.3.4 Скорочення даних

- Зменшення розмірності
- Зменшення кількості
- Стиснення даних

1.4 Функціональна інженерія

У часових рядах дуже часто зустрічається візерунок продажів товарів не з реальних абсолютних значень, а зі зміни цих значень у часі, тому багато сайтів пишуть рекомендації, що потрібно створити функціональність, яка буде орієнтована: у той час; на дату, коли товар буде проданий і ін.

- Функції, пов'язані з датою
- Функції, пов'язані з часом
- Функції перетворення категорій
- Перетворення тексту
- Функції затримки
- Прокручування вікна
- Розширення вікна
- Залежне від домену

Слід зазначити, що Feature engineering дуже пов'язаний з дослідницьким аналізом.

Розглянемо, який вид функцій може бути сформований для часових рядів:

1.4.1 Особливості дати і часу

- Вихідні чи ні
- Літній час чи ні
- Державні свята чи ні
- Чверть року
- Час дня
- Сезон року

1.4.2 Перетворення категоріальних ознак

- Пошук і заміна
- Кодування міток
- Одне гаряче кодування
- Нетипове двійкове кодування
- Контрастне кодування
- Цільове кодування

1.4.3 Перетворення тексту

- Набір слів
- Періодичність термінів і зворотна частота документів
- Вбудовування слів
- Слово в вектори

Для особливостей, які побудовані безпосередньо з аналізу метричних ознак. У часових рядах слід виділяти змінні предиктори, зсувні вікна і розширені вікна.

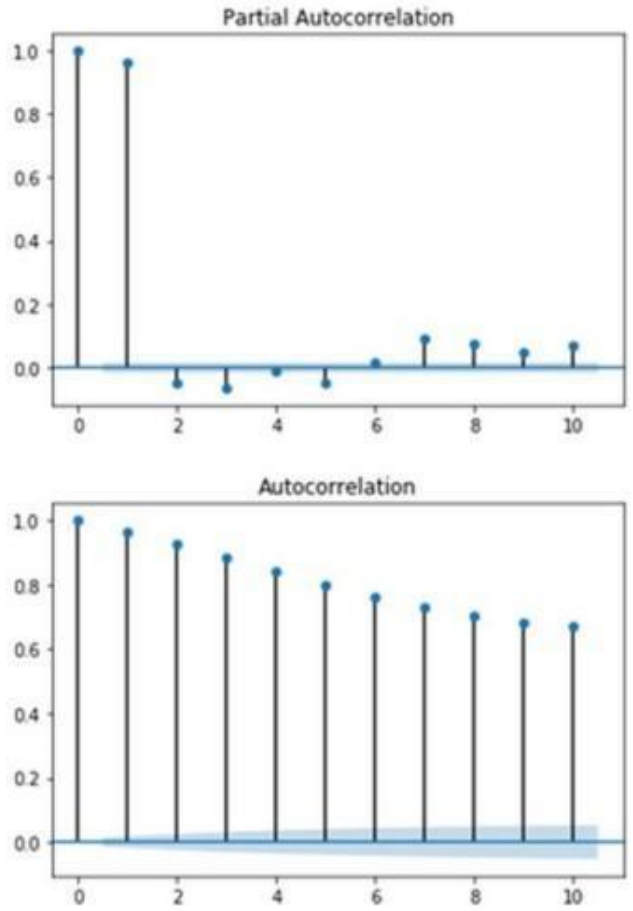


Рис. 1.8. Змінні предиктори

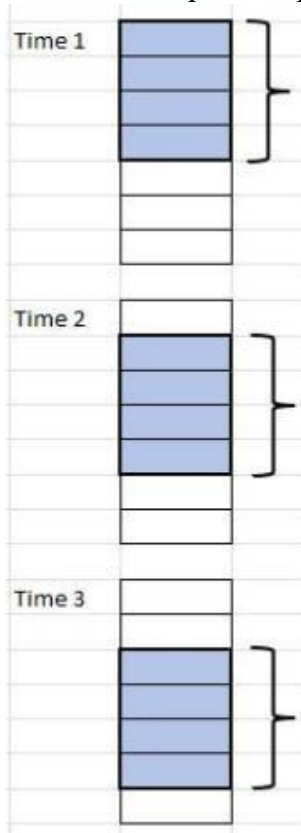


Рис. 1.9. Зсувні вікна

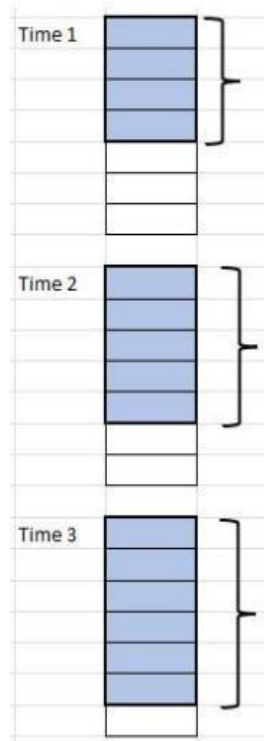


Рис. 1.10. Розширені вікна

Визначення особливостей, що відносяться до предметної області. Наприклад, ми можемо проаналізувати структуру нашої продукції і побачити, як продажі можуть змінюватися відповідно до різних категорій продукції, побачити, як ці продажі поведуться на ринку, спробувати використовувати цю інформацію в DataSet'ах.

1.4.4 Для предметної області

- Створення ознаків затримки з урахуванням комбінації магазин-продукт
- Врахування тенденцій ринку
- Використання знань про продукт
- Створення особливостей відповідно до: сезону, погоди, свят, розташування магазинів тощо
- Використовування будь-яких зовнішніх даних, які можуть вплинути на продаж

1.5 Перевірка моделі та аналіз результатів

Перед тим як перейти до моделювання необхідно визначити заходи, які ми будемо використовувати для аналізу наших моделей. На жаль, задачі класифікації є рідкісними у часових рядах і, наприклад, такі успішні підходи, які передбачають оцінку точності та чутливості не підходять, оскільки ми з більшою ймовірністю маємо справу з регресією у часових рядах. Тому більшість використовуваних показників ближче до регресійних моделей і дозволяють нам обчислити, наскільки близько наш прогноз буде до поточного значення.

1.5.1 Показники точності прогнозу часових рядів

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \text{ — R-Squared}$$

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \text{ — Mean Absolute Error (MAE)}$$

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \text{ — Mean Absolute Percentage Error (MAPE)}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \text{ — Mean Square Error (MSE)}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \text{ — Root Mean Squared Error (RMSE)}$$

$$WMAPE = \frac{\sum_{t=1}^n |A_t - F_t|}{\sum_{t=1}^n |A_t|} \text{ — Weighted Mean Absolute Percentage Error (WMAPE)}$$

Різні моделі застосовуються на основі зсувних або розширених вікон, щоб вивести модель на тренувальний зразок, а вже потім зробити пробний зразок.

1.5.2 Метод перехресної перевірки для часових рядів



Рис. 1.11. Зсунуте вікно



Рис. 1.12. Розширене вікно

1.6 Метод групового урахування аргументів

Метод групового урахування аргументів (МГУА) представляє собою сімейство індуктивних алгоритмів, спрямованих на моделювання математичних даних з численними параметрами. Основний принцип методу полягає в вибіркового використанні моделей, на базі яких формуються складніші математичні моделі.

Розроблений академіком НАНУ О. Г. Івахненком та його науковою групою, МГУА є типовим індуктивним методом моделювання і вважається одним із найбільш ефективних підходів до структурно-параметричної ідентифікації складних об'єктів, процесів і систем на основі обмежених даних спостережень.

В цілому задача ідентифікації полягає у формуванні за даними вибірки деякої множини моделей різної структури

$$\hat{y}_f = f(X, \hat{\theta}_f) \quad (1)$$

і пошуку оптимальної моделі за умовою

$$f^* = \arg \min_{f \in \mathfrak{F}} C(y, f(X, \hat{\theta}_f)), \quad (2)$$

причому оцінки параметрів в (2) для кожної моделі $f \in \mathfrak{F}$ є розв'язком ще однієї екстремальної задачі виду

$$\hat{\theta}_f = \arg \min_{\theta_f \in R^{s_f}} Q(y, X, \theta_f) \quad (3)$$

де s_f називається складністю моделі f і дорівнює кількості ненульових компонентів у моделі виду (3); Q - критерій якості розв'язку задачі параметричної ідентифікації кожної окремої моделі, що генерується в задачі структурної ідентифікації.

МГУА також пропонує широкий спектр підходів до побудови моделей складних систем, таких як системи масового обслуговування, мережі передачі даних, економічні системи та інші. Метод використовується для побудови як статичних, так і динамічних моделей, а також для аналізу і оптимізації їхньої роботи. Одна з основних особливостей МГУА - це можливість автоматичної генерації моделей залежно від потреб користувача. Це дає змогу вибрати найкращу модель з певного набору варіантів, що значно спрощує процес побудови моделі. Також МГУА використовує класи моделей, які базуються на базисних функціях. Це дозволяє швидко і ефективно побудувати модель, використовуючи набір вже підготовлених функцій. Окрім цього, МГУА використовує критерії якості структур моделей. Ці критерії використовуються для оцінки якості моделі та вибору найкращої структури. Це дозволяє побудувати модель оптимальної складності, що дозволяє отримати найкращі результати. У підсумку, МГУА є потужним методом для побудови моделей складних систем, що володіє багатьма перевагами у порівнянні з іншими методами. Він дає можливість автоматично генерувати моделі, вибирати найкращі структури та оптимізувати їхню складність для досягнення найкращих результатів.

Для оцінки якості моделей застосовуються такі зовнішні критерії, як середня абсолютна похибка (MAE), середня квадратична похибка (MSE), коефіцієнт детермінації (R-квадрат), середня абсолютна відносна похибка (Mape) та інші. Для вибору кращої моделі можна порівнювати значення цих критеріїв для різних моделей і обирати ту, яка має найменшу похибку або найвищий коефіцієнт детермінації. Застосування зовнішніх критеріїв дозволяє оцінити якість моделі на підвбірках даних, що не використовувалися при навчанні моделі. Це дозволяє оцінити загальну здатність моделі до узагальнення і прогнозування нових даних. МГУА використовується для вибору оптимальної складності моделі, яка враховує інформативність наявних даних. При застосуванні цього методу, вибірка ділиться на групи, і для кожної групи обирається оптимальна модель з певним рівнем складності. Потім, за допомогою інших критеріїв, проводиться вибір найкращої моделі серед цих оптимальних

моделей, яка відповідає найбільш точним прогнозам реальних даних. Таким чином, застосування зовнішніх критеріїв і МГУА дозволяє об'єктивно порівняти моделі і вибрати ту, яка має найкращу якість і здатність до узагальнення.

Ефективність методу багато разів підтверджувалася розв'язанням безлічі конкретних задач з областей екології, економіки, гідрометеорології тощо. На основі аналогії між задачею побудови моделі за зашумленими експериментальними даними і задачею проходження сигналу через канал з шумом побудовані теорії завадостійкого моделювання. Основний результат цієї теорії полягає в тому, що складність оптимальної прогнозуючої моделі залежить від рівня невизначеності в даних: чим він вище – тим простішою (більш грубою) має бути оптимальна модель (тим менше оцінюваних параметрів).

МГУА є ефективним інструментом структурної ідентифікації моделей з мінімальною дисперсією помилок прогнозування. Ця теорія розроблена і активно використовується як в Україні, так і за її межами. Завдяки методу критичних дисперсій, МГУА дозволяє аналітично вирішувати різноманітні задачі, такі як порівняльний аналіз критеріїв структурної ідентифікації, планування експериментів, аналіз властивостей методів і інше. Ці аналітичні розв'язки можуть використовуватись як для обмежених вибірок, так і у асимптотичних умовах, що робить цей метод ще більш потужним і універсальним. МГУА також досліджує умови вибору оптимальної структури моделі в залежності від рівня шуму, довжини вибірки, вхідних впливів і параметрів об'єкта. Встановлено тісний взаємозв'язок між цими умовами і засобами МГУА, що дозволяє знаходити оптимальну структуру моделі з мінімальною дисперсією помилки прогнозування. Порівняння ефективності МГУА з іншими методами показало, що він є одним з найефективніших методів побудови моделей з мінімальною дисперсією помилки прогнозування. Ця теорія струнко розв'язує актуальні задачі і має широке застосування в практичних дослідженнях і проектах.

Це робить МГУА особливо привабливим і ефективним для використання в задачах штучного інтелекту і м'яких обчислень. Однією з основних переваг

МГУА є його здатність до ефективного розв'язання задач з великою кількістю даних і високою складністю. МГУА може бути успішно застосований для розв'язання задач класифікації, кластеризації, аналізу даних та прогнозування, забезпечуючи високу точність і швидкість обчислень. Одним з ключових принципів МГУА є самоорганізація, що дозволяє мережі адаптуватися до змін в навколишньому середовищі та самостійно побудовувати оптимальну структуру для виконання поставленої задачі. Це забезпечує більш гнучке та ефективне вирішення завдань штучного інтелекту. Іншою важливою особливістю МГУА є можливість «згортати» побудовану мережу у явний математичний вираз, що дає змогу отримати аналітичний результат і спростити подальший аналіз та пошук закономірностей. Це дозволяє зрозуміти логіку та механізми роботи моделі, що є важливим для розуміння та розвитку штучного інтелекту. Усе це робить МГУА потужним інструментом для вирішення різних задач штучного інтелекту і м'яких обчислень. Він знаходить широке застосування в різних сферах, включаючи виробництво, фінанси, медицину, науку та інші. Отже, МГУА можна вважати одним з найсучасніших методів обчислювального інтелекту і м'яких обчислень.

Існують різні заходи, що залучають вчених, дослідників та фахівців з різних країн світу, які зацікавлені в індуктивному моделюванні. Учасники обмінюються досвідом, представляють свої дослідження і результати, дискутують над новими напрямками та викликами в цій галузі. Такі міжнародні заходи є важливим індикатором популярності МГУА, адже вони показують, що існує активна та зацікавлена спільнота вчених, що займаються дослідженнями та розвитком індуктивного моделювання. Крім того, участь у таких заходах дозволяє МГУА встановлювати контакти зі співробітниками з інших університетів та наукових центрів, обмінюватися ідеями та підтримувати активну наукову співпрацю. Такі міжнародні форуми також позитивно впливають на репутацію МГУА як провідного центру в галузі індуктивного моделювання. Вони свідчать про визнання іншими науковими громадами важливості та значущості досліджень, які проводяться у навчальному закладі. Загалом, міжнародні форуми з індуктивного моделювання, організовані МГУА, є підтвердженням його популярності та впливу у цій галузі. Вони дозволяють

спрямувати увагу на досягнення учених, розвиток нових технологій та інноваційну роботу, яка відбувається у навчальному закладі.

Індуктивне моделювання, засноване на принципах самоорганізації, є підходом, який розвивається протягом 40 років. Воно ефективно використовується в багатьох областях і є основою для таких технологій аналізу даних, як поліноміальні нейронні мережі, адаптивні та статистичні нейронні мережі.

В останні роки для побудови моделей на основі даних також використовуються еволюційні й генетичні алгоритми, ідея активних нейронів і багаторівнева самоорганізація. Ці нові розробки доповнюють та розширюють існуючі підходи до індуктивного моделювання і надають нові можливості для аналізу та прогнозування даних.

Наприклад, генетичні алгоритми дозволяють побудувати моделі, які здатні ефективно вирішувати складні задачі оптимізації. Еволюційні алгоритми враховують принципи природного відбору, що дозволяє знаходити оптимальні рішення на основі ітеративного процесу. Ідея активних нейронів полягає в тому, що кожен нейрон має власну активність і може регулювати свою активність залежно від отримуваної інформації. Це дозволяє покращити результати моделювання і збільшити його адаптивність до змінних умов.

Багаторівнева самоорганізація передбачає побудову моделі зі складових, які організовані у формі ієрархії. Це дозволяє розмежувати і розуміти складні структури даних і знаходити залежності між ними.

Використання цих інноваційних ідей у індуктивному моделюванні дозволяє покращити якість аналізу даних, знайти нові залежності і зробити більш точні прогнози. Вони також допомагають створювати більш ефективні технології аналізу даних, які можуть бути застосовані в різних галузях, включаючи бізнес, медицину та науку.

1.7 Висновки до розділу 1

Завдання якомога більш точного та оперативного прогнозування продажів є однією з найважливіших для багатьох компаній (особливо для представників сектора роздрібної торгівлі). А інвестиції в цей напрямок, якщо буде обраний правильний підхід, окупляться дуже швидко.

Сучасні пакети аналізу пропонують різноманітні методи аналізу та прогнозування часових рядів продажів. Однак при широкому асортименті виробів підхід погляду на кожен ряд окремо, виконання їх детального аналізу і проекційної побудови стає непрактичним.

Тому застосовуються й інші підходи до цієї задачі. Одним з них є використання методів сегментації, які часто можуть значно зменшити розмір проблеми.

Більш перспективним і надійним підходом виявляється використання програмних рішень, які автоматизують процес прогнозування і роблять оптимальним використання його ресурсів.

РОЗДІЛ 2 СПЕЦІАЛЬНИЙ

2.1 Постановка задачі

Спробуємо спрогнозувати на весь 2022 рік прибуток на прикладі магазину одягу «LC Waikiki», який саме займається роздрібними продажами.

Для цієї роботи буде використане інтерактивне хмарне середовище Google Colaboratory та програмне забезпечення GMDH Shell.

Дані, які будуть використовуватись у розробці прогнозу, зазначені у додатку В.

2.1.1 Завантаження даних та бібліотек

Завантажуємо необхідні бібліотеки та функції.

```
import matplotlib.pyplot as plt
%matplotlib inline

import pandas as pd
import numpy as np
from pandas.plotting import scatter_matrix

from adtk.data import validate_series
from adtk.visualization import plot
from adtk.detector import ThresholdAD
from adtk.detector import OutlierDetector

from sklearn.neighbors import LocalOutlierFactor

from scipy.stats import variation
import seaborn as sns

import matplotlib.pyplot as plt

import statsmodels.api as sm
from statsmodels.tsa.seasonal
import seasonal_decompose
from statsmodels.tsa.api import ExponentialSmoothing

from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_pacf, plot_acf
from statsmodels.stats.outliers_influence
import variance_inflation_factor
```

```

from itertools import compress,
product from pmdarima import auto_arima
import warnings
warnings.filterwarnings("ignore")

def sesonal(data, s):
plt.figure(figsize=(19,8), dpi= 80)
for i, y in enumerate(data.index.year.unique()):
    plt.plot(list(range(1,len(data[data.index.year==y])+1)), data[data.index.year==y][data.columns[0]].values, label=y)
plt.title("Сезонність за періодами")
plt.legend(loc="best")
plt.show()

def metrics(real, forecast):

if type(real)==pd.core.frame.DataFrame:
real=real[real.columns[0]].values

print("Тест на стаціонарність:")
dfstest = adfuller(real-forecast, autolag='AIC')
print("\tT-статистика = {:.3f}".format(dfstest[0]))
print("\tP-значення = {:.3f}".format(dfstest[1]))
print("Критичні значення :")
for k, v in dfstest[4].items():
print("\t{ }: { } - Дані { } стаціонарні з ймовірністю { }% відсотків".format(k, v, "ні" if v<dfstest[0] else "", 100-int(k[:-1])))

forecast=np.array(forecast)
print('MAD:', round(abs(real-forecast).mean(),4))
print('MSE:', round(((real-forecast)**2).mean(),4))
print('MAPE:', round((abs(real-forecast)/real).mean(),4))
print('MPE:', round(((real-forecast)/real).mean(),4))
print('Стандартна похибка:', round(((real-forecast)**2).mean()*0.5,4))

def metrics_short(real, forecast): real=np.array(real[real.columns[0]].values) forecast=np.array(forecast)
print('MAD:', round(abs(real-forecast).mean(),4))
print('MSE:', round(((real-forecast)**2).mean(),4))
print('MAPE:', round((abs(real-forecast)/real).mean(),4))
print('MPE:', round(((real-forecast)/real).mean(),4))
print('Стандартна похибка:', round(((real-forecast)**2).mean()*0.5,4))

```

```

def h_map(data, level):
    corr = data.corr() plt.figure(figsize=(14, 14))
sns.heatmap(corr[(corr >= level) | (corr <= -level)],
cmap="RdBu_r", vmax=1.0, vmin=-1.0, linewidths=0.1, annot=True,
annot_kws={"size": 8}, square=True)
    plt.show()

def combinations(items):
    return list( set(compress(items,mask)) for mask in product(*[[0,1]]*len(items)) )
    def get_factors(data, Y, columns):
    spisCol=combinations(columns)
    print('Кількість комбінацій', len(spisCol))
    data=sm.add_constant(data)
    arr_res=[]

    for c in spisCol:
    perem=list(c)
    flag=True

    if len(perem)==0: continue

    if not('const' in c):
    perem.append('const')

    if len(perem)>1:
        vif = [variance_inflation_factor(data[perem].values, i) for i in range(data[perem].shape[1])]
    else:
    vif=[]

    for vv in vif:
    if vv>1000:
    flag=False

    reg = sm.OLS(Y, data[perem])
    res=reg.fit()

    for val in res.tvalues:
    if val<2 and val>2:
        flag=False
        break
    for val in res.pvalues:
    if val>0.05:
        flag=False
        break
    if flag:

```

```

re=np.array(res.fittedvalues.copy())
MSE=((np.array(Y)-re)**2).sum()/len(re)

MAPE=(abs((np.array(Y)-re)/np.array(Y))).sum()/len(re)
arr_res.append([round(MSE,4), res.rsquared, perem])
arr_res.sort()
df_model=pd.DataFrame(arr_res, columns=['MSE', 'r2', 'Змінні'])
print('Результати перебору в порядку збільшення MSE:')
print(df_model)
return df_model

```

Завантажимо файл з даними:

```
df=pd.read_excel("sample_dataset.xlsx", index_col='Xm', sheet_name='Data')
```

```
df.head()
```

	Y
Xm	
2017-01-01	3007542
2017-02-01	1699306
2017-03-01	1235783
2017-04-01	1005879
2017-05-01	1529041

```
df.tail()
```

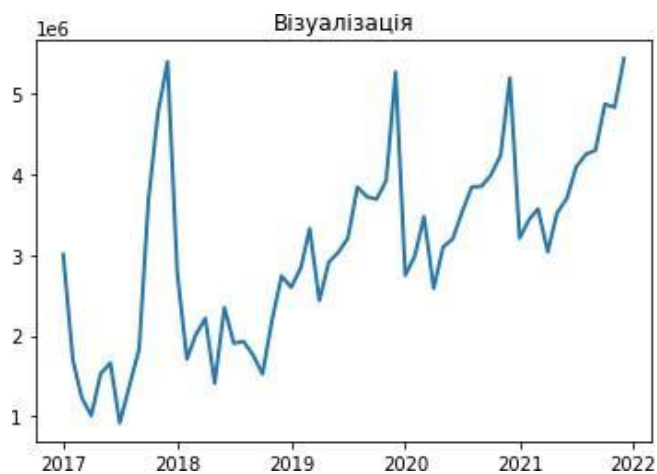
	Y
Xm	
2021-08-01	4249313
2021-09-01	4299709
2021-10-01	4872711
2021-11-01	4831135
2021-12-01	5436279

Щоб перевірити ряд на відсутність пропусків, виконаємо його валідацію.

```
df = validate_series(df)
print(df)
```

Xm		2019-01-01	2599300		
2017-01-01	3007542	2019-02-01	2843485		
2017-02-01	1699306	2019-03-01	3328551		
2017-03-01	1235783	2019-04-01	2435684		
2017-04-01	1005879	2019-05-01	2909290		
2017-05-01	1529041	2019-06-01	3035527		
2017-06-01	1660054	2019-07-01	3206635		
2017-07-01	916857	2019-08-01	3842120		
2017-08-01	1376223	2019-09-01	3721218		
2017-09-01	1833750	2019-10-01	3692009		
2017-10-01	3685075	2019-11-01	3925217		
2017-11-01	4783007	2019-12-01	5270964		
2017-12-01	5398573	2020-01-01	2749207		
2018-01-01	2785230	2020-02-01	2993479	2021-01-01	3209207
2018-02-01	1707580	2020-03-01	3477284	2021-02-01	3443479
2018-03-01	2009750	2020-04-01	2586093	2021-03-01	3572518
2018-04-01	2212900	2020-05-01	3100257	2021-04-01	3039082
2018-05-01	1410825	2020-06-01	3200753	2021-05-01	3523724
2018-06-01	2348528	2020-07-01	3535891	2021-06-01	3700062
2018-07-01	1905324	2020-08-01	3844218	2021-07-01	4092781
2018-08-01	1928276	2020-09-01	3852107	2021-08-01	4249313
2018-09-01	1757532	2020-10-01	3991024	2021-09-01	4299709
2018-10-01	1525121	2020-11-01	4236044	2021-10-01	4872711
2018-11-01	2200500	2020-12-01	5193643	2021-11-01	4831135
2018-12-01	2738434	2021-01-01	3209207	2021-12-01	5436279

```
plt.plot(df, linewidth=2)
plt.title("Візуалізація")
plt.show()
```



2.1.2 Виявлення та коригування аномалій

Наступна команда дає змогу вивчити основні статистики ряду.

```
df.describe()
```

	Y
count	6.000000e+01
mean	3.075018e+06
std	1.162257e+06
min	9.168570e+05
25%	2.152812e+06
50%	3.069670e+06
75%	3.842644e+06
max	5.436279e+06

Використовуючи ці дані, подивимося на дані та можливі викиди.

```
threshold_ad = ThresholdAD(high=5000000, low=1200000)
```

```
anomalies = threshold_ad.detect(df)
```

```
plot(df, anomaly=anomalies, ts_linewidth=1, ts_markersize=3, anomaly_markersize=5, anomaly_color='red', anomaly_tag="marker" );
```

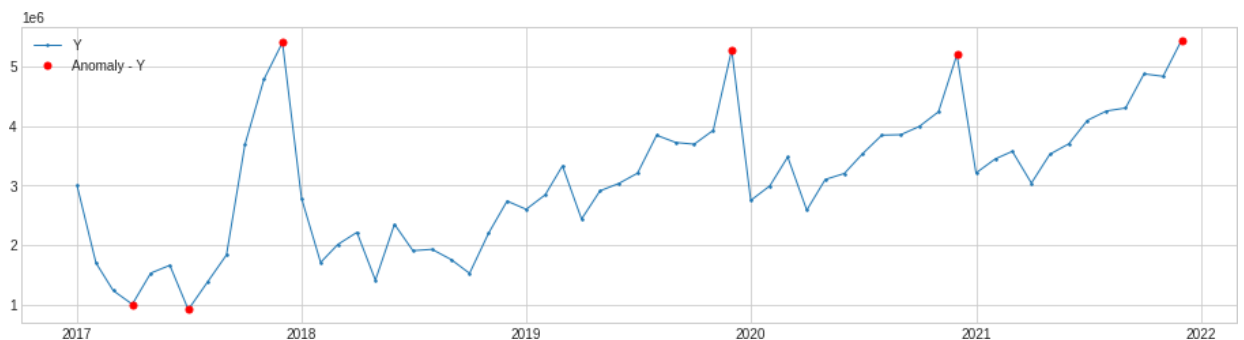


Рис. 2.1. Аномалії

Коригуючи параметри `high` і `low`, можна коригувати область детекції аномалій. Самі аномалії підсвічені червоними крапками. Щоб побачити, які дані потрапили до аномалії, достатньо виконати команду:

```
anomalies[anomalies.Y]
```

Y	
Xm	
2017-04-01	True
2017-07-01	True
2017-12-01	True
2019-12-01	True
2020-12-01	True
2021-12-01	True

```

outlier_detector = OutlierDetector(LocalOutlierFactor(contamination=0.06))
anomalies = outlier_detector.fit_detect(df)
plot(df, anomaly=anomalies, ts_linewidth=2, anomaly_color='red', anomaly_alpha=0.3, curve_group='all');

```

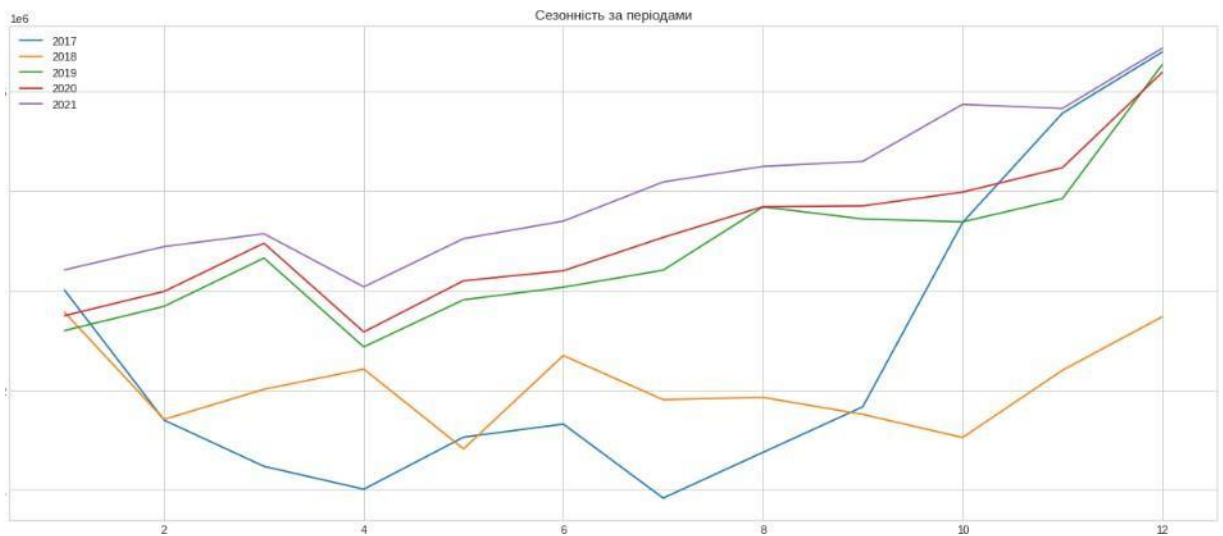


Аномалії підсвічені червоним. Змінюючи параметр `contamination`, можна міняти чутливість моделі. Чим більше значення, тим більш підозріло вона ставитиметься до аномалій.

Ще один тест на аномальність – це оцінка наявності сезонних компонентів.

Побудуємо суміщені графіки за роками.


```
seasonal(df,12)
```



2.1.3 Декомпозиція на основі тренду та сезонності

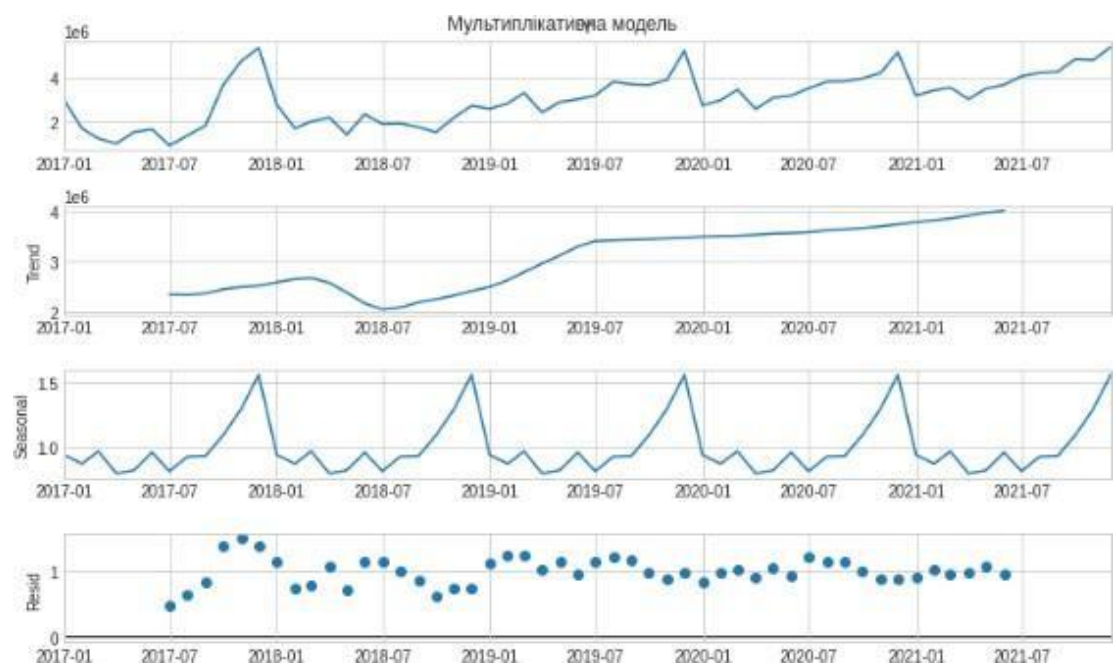
Виконаємо розкладання ряду за мультиплікативною моделлю та отримаємо результати.

Щоб отримати значення тренду, треба виконати команду:

```
result_M = seasonal_decompose(df.Y, model='multiplicative')
```

```
plt.rcParams.update({'figure.figsize': (10,6)})
```

```
result_M.plot().suptitle('Мультиплікативна модель')
```

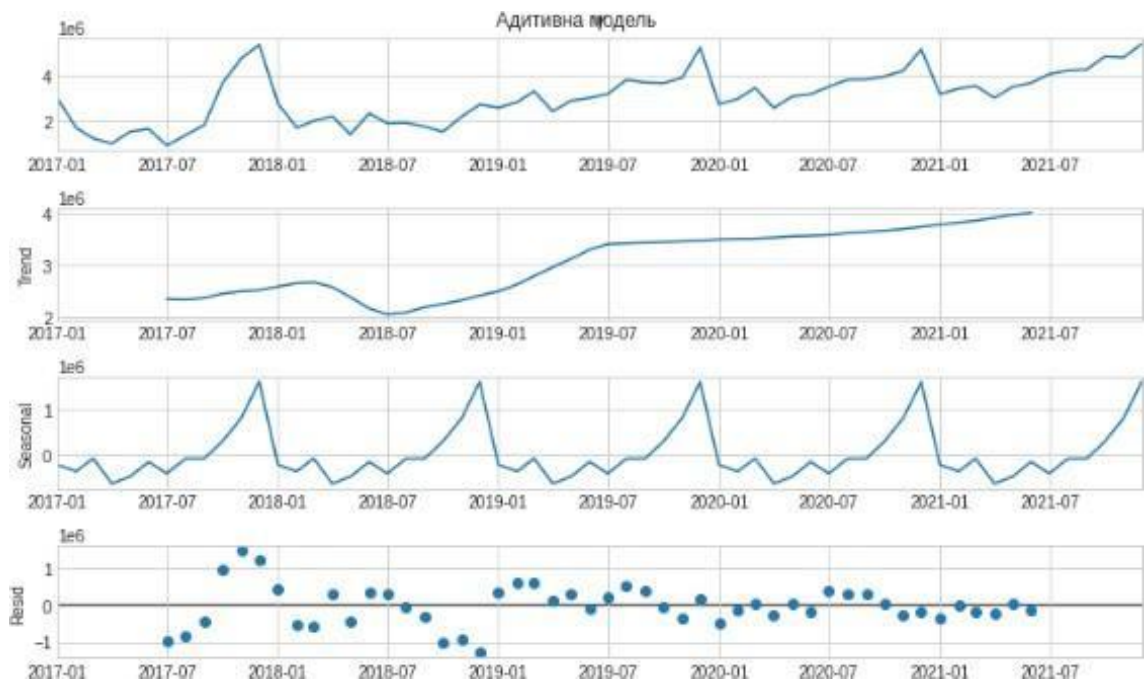


Виконаємо адитивне розкладання ряду.

```
result_A = seasonal_decompose(df.Y, model='additive')
```

```
plt.rcParams.update({'figure.figsize': (10,6)})
```

```
result_A.plot().suptitle('Адитивна модель')
```



Бачимо, що ряд нестационарний і з явно вираженою сезонною компонентою

2.1.4 Тренувальні та тестові набори

Щоб мати можливість якісніше оцінювати модель, розділимо дані на тренувальні та тестові. Частину даних ми «сховаємо» від моделі і намагатимемося їх передбачити.

```
train=df['2017':'2020']
```

```
train.head()
```

	Y
Xm	
2017-01-01	3007542
2017-02-01	1699306
2017-03-01	1235783
2017-04-01	1005879
2017-05-01	1529041

```
test=df['2021']
```

```
test.tail()
```

	Y
Xm	
2021-08-01	4249313
2021-09-01	4299709
2021-10-01	4872711
2021-11-01	4831135
2021-12-01	5436279

2.2 Модель Хольта-Вінтерса

Багато продуктів мають тенденцію зростання чи падіння продажів, особливо коли вони виробляються вперше чи коли з'являються конкуруючі товари.

Для деяких продуктів суттєві сезонні зміни рівня продажів, тому для прогнозу продажу товару доцільно враховувати конкретний характер тенденції та сезонних коливань. На основі моделі Хольта Вінтерс створив свою прогностичну модель, яка враховує експоненційний тренд та адитивну сезонність.

```
fit1 = ExponentialSmoothing(train, seasonal_periods=12, trend='add',
seasonal='mul').fit()
```

```
fit1.params
```

```
{'damping_trend': nan,
'initial_level': 2279743.088888888,
'initial_seasons': array([0.97523901, 0.86554005, 0.98490285, 0.80977482, 0.80377435,
0.97439019, 0.76032526, 0.88582788, 0.89403628, 1.0926812 ,
1.34320743, 1.61030066]),
'initial_trend': 38551.35050505065,
'lamda': None,
'remove_bias': False,
'smoothing_level': 0.995,
'smoothing_seasonal': 0.0001,
'smoothing_trend': 0.0001,
'use_boxcox': False}
```

В даному випадку реалізована класична модель з адитивним трендом та мультиплікативною сезонною компонентою.

Ми можемо отримати розрахункові значення для моделі та розрахувати метрики точності.

```
fit1.fittedvalues
```

Xm		2019-01-01	1.695834e+06
2017-01-01	2.260891e+06	2019-02-01	2.336027e+06
2017-02-01	2.699361e+06	2019-03-01	3.270804e+06
2017-03-01	1.977271e+06	2019-04-01	2.767853e+06
2017-04-01	1.050219e+06	2019-05-01	2.450206e+06
2017-05-01	1.029536e+06	2019-06-01	3.561679e+06
2017-06-01	1.888089e+06	2019-07-01	2.399869e+06
2017-07-01	1.325495e+06	2019-08-01	3.765596e+06
2017-08-01	1.104609e+06	2019-09-01	3.911906e+06
2017-09-01	1.421979e+06	2019-10-01	4.591451e+06
2017-10-01	2.280734e+06	2019-11-01	4.595727e+06
2017-11-01	4.573222e+06	2019-12-01	4.771608e+06
2017-12-01	5.795055e+06	2020-01-01	3.228521e+06
2018-01-01	3.308479e+06	2020-02-01	2.475094e+06
2018-02-01	2.507458e+06	2020-03-01	3.441330e+06
2018-03-01	1.985500e+06	2020-04-01	2.890172e+06
2018-04-01	1.683498e+06	2020-05-01	2.599416e+06
2018-05-01	2.224965e+06	2020-06-01	3.792783e+06
2018-06-01	1.752579e+06	2020-07-01	2.529146e+06
2018-07-01	1.859485e+06	2020-08-01	4.147918e+06
2018-08-01	2.253787e+06	2020-09-01	3.915874e+06
2018-09-01	1.982185e+06	2020-10-01	4.750544e+06
2018-10-01	2.191483e+06	2020-11-01	4.962433e+06
2018-11-01	1.930335e+06	2020-12-01	5.144659e+06
2018-12-01	2.698247e+06		

Freq: MS, dtype: float64

Розрахуємо прогноз на 24 місяці уперед.

```
fit1.forecast(24)
```

```

2021-01-01    3.182924e+06
2021-02-01    2.857952e+06
2021-03-01    3.289936e+06
2021-04-01    2.736192e+06
2021-05-01    2.746955e+06
2021-06-01    3.367328e+06
2021-07-01    2.656941e+06
2021-08-01    3.129452e+06
2021-09-01    3.192857e+06
2021-10-01    3.944239e+06
2021-11-01    4.900191e+06
2021-12-01    5.936607e+06
2022-01-01    3.633053e+06
2022-02-01    3.257416e+06
2022-03-01    3.744485e+06
2022-04-01    3.109931e+06
2022-05-01    3.117941e+06
2022-06-01    3.817037e+06
2022-07-01    3.007871e+06
2022-08-01    3.538292e+06
2022-09-01    3.605488e+06
2022-10-01    4.448544e+06
2022-11-01    5.520118e+06
2022-12-01    6.679817e+06
Freq: MS, dtype: float64

```

```
metrics(test, fit1.forecast(len(test)))
```

```

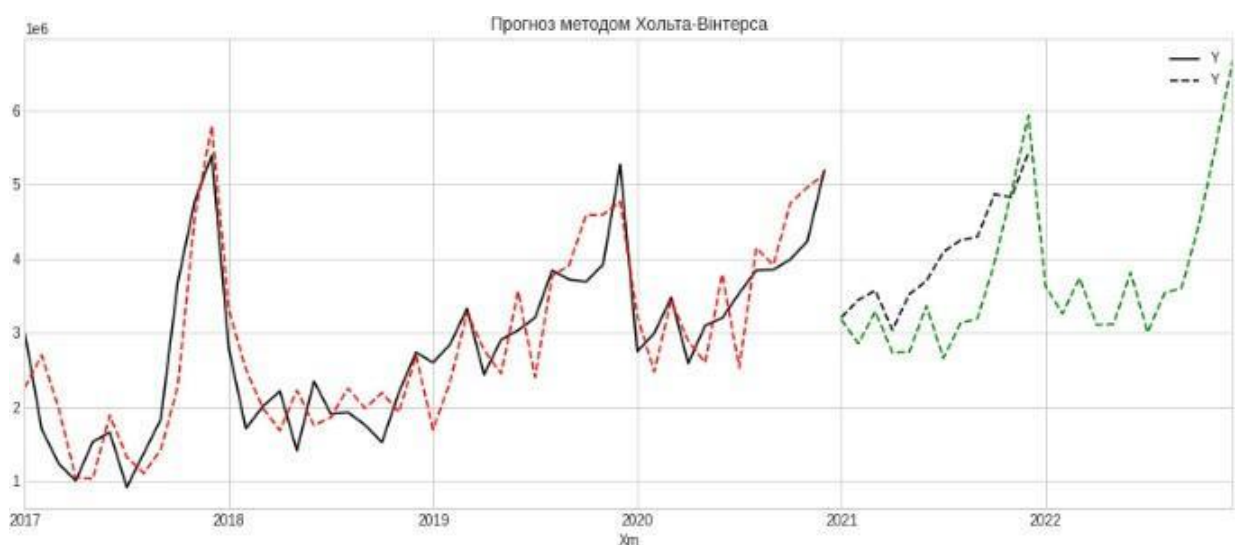
Тест на стаціонарність:
  Т-статистика = -3.907
  Р-значення = 0.002
Критичні значення :
  1%: -4.6651863281249994 - Дані ні стаціонарні з ймовірністю 99% відсотків
  5%: -3.3671868750000002 - Дані стаціонарні з ймовірністю 95% відсотків
 10%: -2.802960625 - Дані стаціонарні з ймовірністю 90% відсотків
MAD: 622266.2588
MSE: 573934550729.9542
MAPE: 0.153
MPE: 0.1353
Стандартна похибка: 757584.6822

```

```

ax = train.plot(figsize=(15,6), color='black', title="Прогноз методом Хольта-
Вінгерса")
test.plot(ax=ax, style='--', color='black')
fit1.fittedvalues.plot(ax=ax, style='-', color='red')
fit1.forecast(24).plot(ax=ax, style='--',
color='green')
plt.show()

```



Чорною пунктирною лінією відображено тестовий період відносно спрогнозованого.

2.3 Модель SARIMA

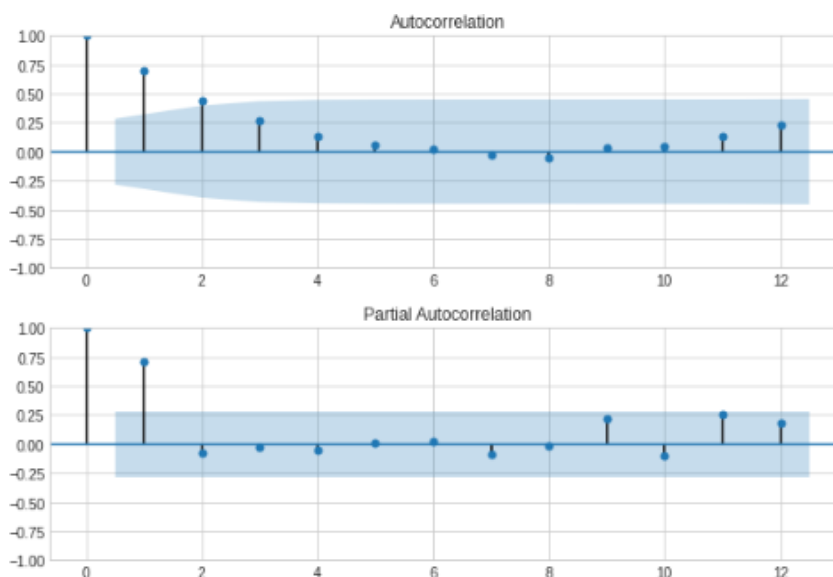
Сезонне авторегресійне інтегроване середнє ковзне, SARIMA або Seasonal ARIMA, є розширенням ARIMA, яке явно підтримує одновимірні дані часових рядів із сезонним компонентом.

Він додає три нові гіперпараметри для вказівки авторегресії (AR), різниці (I) та ковзного середнього (MA) для сезонної складової ряду, а також додатковий параметр для періоду сезонності.

Сезонна модель ARIMA формується шляхом включення додаткових сезонних термінів до ARIMA. Сезонна частина моделі складається з термінів, які дуже схожі на несезонні компоненти моделі, але включають зворотні зрушення сезонного періоду.

Підбір параметрів моделі зазвичай проводять за допомогою вивчення графіка автокореляції.

```
plt.rcParams.update({'figure.figsize':
(10,3)})
plot_acf(train.Y, lags=12)
plot_pacf(train.Y,
lags=12)
plt.show()
```



Підберемо параметри моделі.

```
model = auto_arima(train, seasonal=True, m=12, trace=True, suppress_warnings=True, error_action='ignore', stepwise=True,
t='tren d')
```

```
model
```

```
Performing stepwise search to minimize aic
ARIMA(2,1,2)(1,0,1)[12] intercept : AIC=1420.177, Time=0.97 sec
ARIMA(0,1,0)(0,0,0)[12] intercept : AIC=1417.105, Time=0.01 sec
ARIMA(1,1,0)(1,0,0)[12] intercept : AIC=1419.830, Time=0.06 sec
ARIMA(0,1,1)(0,0,1)[12] intercept : AIC=1419.710, Time=0.06 sec
ARIMA(0,1,0)(0,0,0)[12] : AIC=1415.267, Time=0.01 sec
ARIMA(0,1,0)(1,0,0)[12] intercept : AIC=1418.860, Time=0.05 sec
ARIMA(0,1,0)(0,0,1)[12] intercept : AIC=1418.933, Time=0.05 sec
ARIMA(0,1,0)(1,0,1)[12] intercept : AIC=1417.711, Time=0.28 sec
ARIMA(1,1,0)(0,0,0)[12] intercept : AIC=1418.162, Time=0.03 sec
ARIMA(0,1,1)(0,0,0)[12] intercept : AIC=1417.928, Time=0.03 sec
ARIMA(1,1,1)(0,0,0)[12] intercept : AIC=1419.366, Time=0.09 sec

Best model: ARIMA(0,1,0)(0,0,0)[12]
Total fit time: 1.672 seconds
ARIMA(order=(0, 1, 0), scoring_args={}, seasonal_order=(0, 0, 0, 12),
suppress_warnings=True, with_intercept=False)
```

Підберемо та розрахуємо найточнішу модель.

```
mod = sm.tsa.statespace.SARIMAX(train,
                                order=(2, 1, 2),
                                seasonal_order=(1, 1, 1, 12))
results = mod.fit()
```

```
print(results.summary())
```

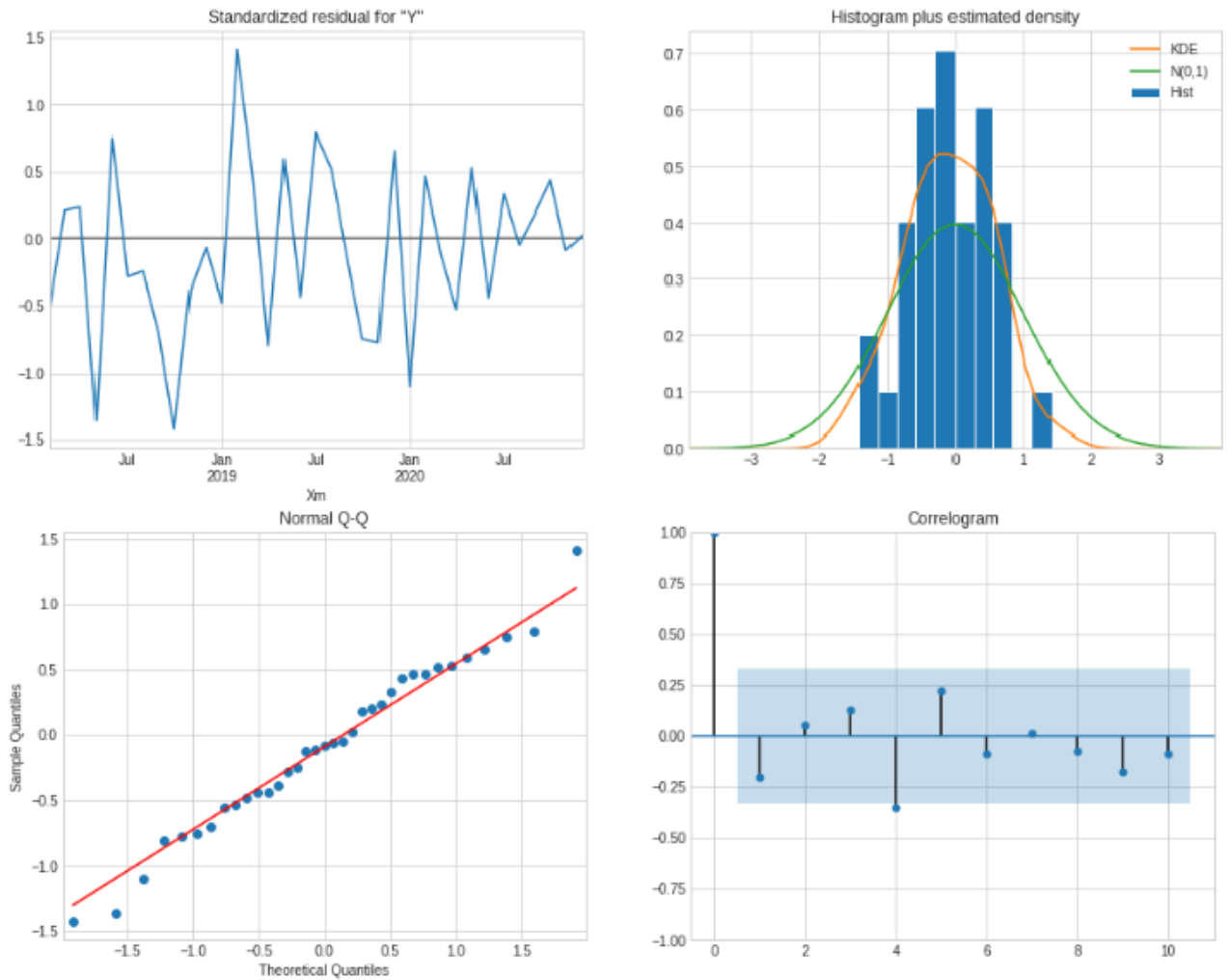
```

SARIMAX Results
=====
Dep. Variable:                Y      No. Observations:                48
Model:                SARIMAX(2, 1, 2)x(1, 1, [1], 12)  Log Likelihood                -520.230
Date:                Tue, 07 Jun 2022  AIC                1054.460
Time:                15:47:50      BIC                1065.348
Sample:                01-01-2017  HQIC                1058.219
                                - 12-01-2020

Covariance Type:                opg
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----
ar.L1         -0.1606     0.781     -0.206     0.837     -1.691     1.369
ar.L2          0.7119     0.830     0.857     0.391     -0.915     2.339
ma.L1          0.0338     0.670     0.050     0.960     -1.279     1.347
ma.L2         -0.8828     0.794    -1.112     0.266     -2.438     0.673
ar.S.L12      -0.5635     0.350    -1.610     0.107     -1.250     0.123
ma.S.L12       0.1360     0.556     0.245     0.807     -0.954     1.226
sigma2       8.556e+11  1.34e-13  6.37e+24  0.000     8.56e+11  8.56e+11
=====
Ljung-Box (L1) (Q):                1.57  Jarque-Bera (JB):                0.18
Prob(Q):                0.21  Prob(JB):                0.92
Heteroskedasticity (H):            0.43  Skew:                -0.07
Prob(H) (two-sided):            0.16  Kurtosis:                2.68
=====
```

Побудуємо діагностичні графіки для отриманої моделі.

```
results.plot_diagnostics(figsize=(15,
12))
plt.show()
```



Оцінимо точність нашої моделі за стандартними метриками. Але врахуємо, що модель не будує прогноз першого періоду. Це можна побачити, вивівши результати та не ставлячи початкову дату.

```
predict=results.get_prediction()
predict.predicted_mean[:10]
```

```
Xm
2017-01-01    0.000000e+00
2017-02-01    2.757403e+06
2017-03-01    1.523758e+06
2017-04-01    1.237792e+06
2017-05-01    9.537903e+05
2017-06-01    1.514040e+06
2017-07-01    1.560597e+06
2017-08-01    9.894447e+05
2017-09-01    1.371870e+06
2017-10-01    1.817817e+06
Freq: MS, Name: predicted_mean, dtype: float64
```



```
predict=results.get_prediction(start=pd.to_datetime('2017-02-01'))
```

```
metrics(train['2017-02-01:'], predict.predicted_mean)
```

```
Тест на стаціонарність:
  Т-статистика = -4.818
  Р-значення = 0.000
Критичні значення :
  1%: -3.5885733964124715 - Дані стаціонарні з ймовірністю 99% відсотків
  5%: -2.929885661157025 - Дані стаціонарні з ймовірністю 95% відсотків
  10%: -2.6031845661157025 - Дані стаціонарні з ймовірністю 90% відсотків
MAD: 611857.5733
MSE: 763619083144.1985
MAPE: 0.2566
MPE: -0.077
Стандартна похибка: 873853.01
```

Порівняємо результати з тестовими даними та побудуємо графік.

```
metrics(test, results.forecast(len(test)))
```

```
Тест на стаціонарність:
  Т-статистика = -6.942
  Р-значення = 0.000
Критичні значення :
  1%: -4.9386902332361515 - Дані стаціонарні з ймовірністю 99% відсотків
  5%: -3.477582857142857 - Дані стаціонарні з ймовірністю 95% відсотків
  10%: -2.8438679591836733 - Дані стаціонарні з ймовірністю 90% відсотків
MAD: 645585.7025
MSE: 467094455299.4019
MAPE: 0.1662
MPE: -0.1662
Стандартна похибка: 683443.0886
```

```
ax = train.plot(figsize=(15,6), color='black', title="Прогноз методом SARIMA")
```

```
test.plot(ax=ax, style='--', color='black')
```

```
predict.predicted_mean.plot(ax=ax, style='--', color='red')
```

```
results.forecast(24).plot(ax=ax, style='--', color='green')
```

```
plt.show()
```



Отже, модель виявилась адекватною, тобто, її рух сходиться із фактичними значеннями.

2.4 Модель лінійної регресії

Лінійна регресія - модель залежності змінної x від однієї або кількох інших змінних (чинників, регресорів, незалежних змінних) з лінійною функцією залежності.

Лінійна регресія відноситься до завдання визначення «лінії найкращої відповідності» через набір точок даних і стала простим попередником нелінійних методів, які використовують для навчання нейронних мереж.

Прочитаємо дані для регресії.

```
data = pd.read_csv('sample_dataset.csv', header = 0, sep = ',')
```

```
data['Xm'] = pd.to_datetime(data['Xm'])
```

```
data = data.drop(['Unnamed: 2'], axis = 1)
```

```
data = data.iloc[:-1]
```

```
data = data.set_index(['Xm'])
```

```
data
```

Xm		Y	
2017-01-01	3007542.0	2019-01-01 2599300.0	
2017-02-01	1699306.0	2019-02-01 2843485.0	
2017-03-01	1235783.0	2019-03-01 3328551.0	
2017-04-01	1005879.0	2019-04-01 2435684.0	
2017-05-01	1529041.0	2019-05-01 2909290.0	
2017-06-01	1680054.0	2019-06-01 3035527.0	
2017-07-01	916857.0	2019-07-01 3206635.0	
2017-08-01	1376223.0	2019-08-01 3842120.0	
2017-09-01	1833750.0	2019-09-01 3721218.0	
2017-10-01	3685075.0	2019-10-01 3692009.0	
2017-11-01	4783007.0	2019-11-01 3925217.0	
2017-12-01	5396573.0	2019-12-01 5270964.0	
2018-01-01	2785230.0	2020-01-01 2749207.0	2021-01-01 3209207.0
2018-02-01	1707580.0	2020-02-01 2993479.0	2021-02-01 3443479.0
2018-03-01	2009750.0	2020-03-01 3477284.0	2021-03-01 3572518.0
2018-04-01	2212900.0	2020-04-01 2586093.0	2021-04-01 3039082.0
2018-05-01	1410825.0	2020-05-01 3100257.0	2021-05-01 3523724.0
2018-06-01	2348528.0	2020-06-01 3200753.0	2021-06-01 3700082.0
2018-07-01	1905324.0	2020-07-01 3535891.0	2021-07-01 4092781.0
2018-08-01	1928276.0	2020-08-01 3844218.0	2021-08-01 4249313.0
2018-09-01	1757532.0	2020-09-01 3852107.0	2021-09-01 4299709.0
2018-10-01	1525121.0	2020-10-01 3991024.0	2021-10-01 4872711.0
2018-11-01	2200500.0	2020-11-01 4236044.0	2021-11-01 4831135.0
2018-12-01	2738434.0	2020-12-01 5193643.0	2021-12-01 5436279.0

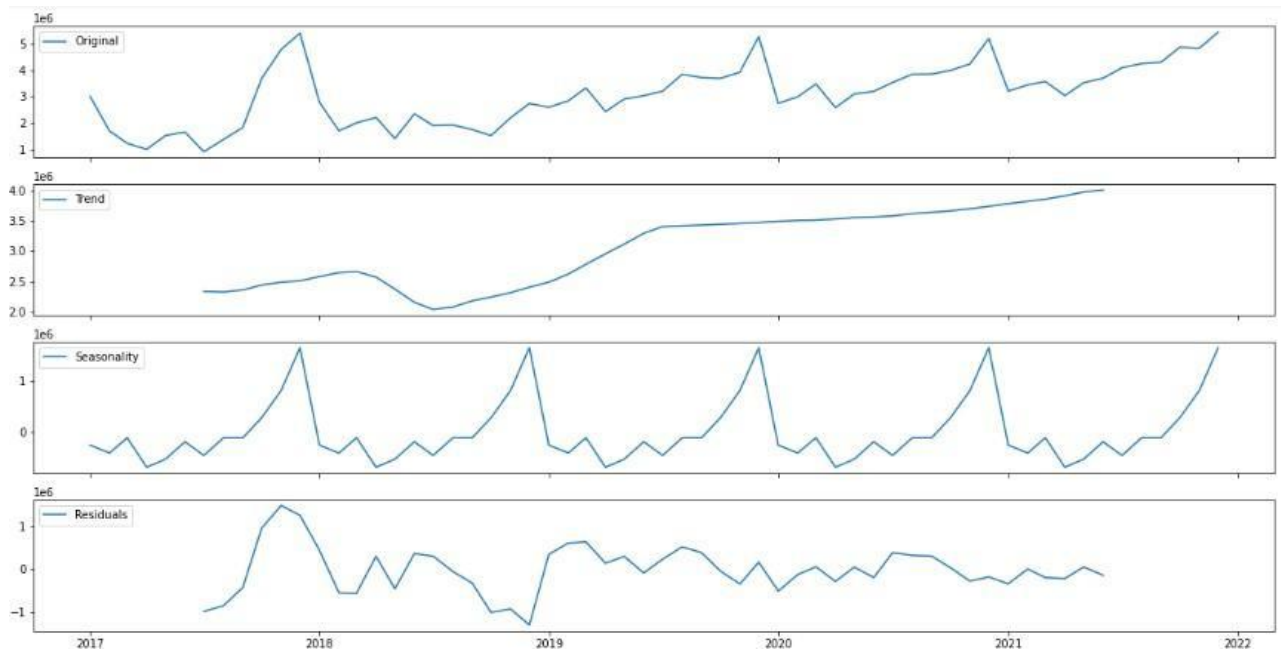
```

result = seasonal_decompose(data, model='additive')

trend_estimate = result.trend
seasonal_estimate = result.seasonal
residual_estimate = result.resid

fig, axes = pyplot.subplots(4, 1, sharex=True, sharey=False)
fig.set_figheight(10)
fig.set_figwidth(20)
axes[0].plot(data, label='Original')
axes[0].legend(loc='upper left');
axes[1].plot(trend_estimate, label='Trend')
axes[1].legend(loc='upper left');
axes[2].plot(seasonal_estimate, label='Seasonality')
axes[2].legend(loc='upper left');
axes[3].plot(residual_estimate, label='Residuals')
axes[3].legend(loc='upper left');

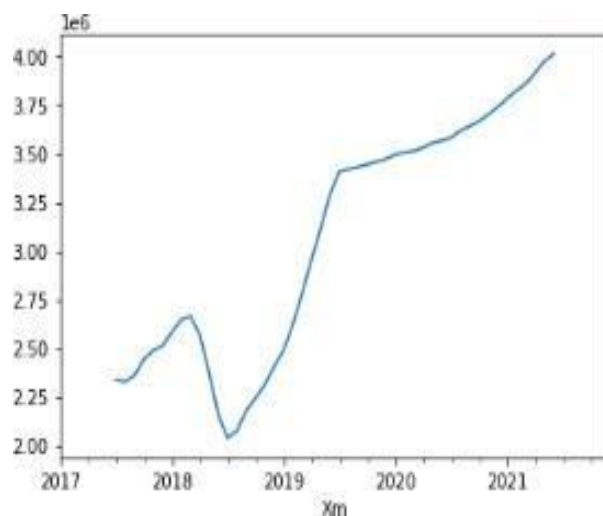
```



```

d_estimate.interpolate(method='polynomial', order=2, limit_direction='both', axis=0).plot()

```



```

trend_estimate = trend_estimate.dropna()

trend = pd.DataFrame(trend_estimate, columns=['trend']).reset_index(drop = True)
import sklearn

from sklearn import linear_model
x=np.array(trend.index)
y = trend.values
body_reg = linear_model.LinearRegression()
x_values = x.reshape(-1, 1)
y_values = y.reshape(-1, 1)
body_reg.fit(x_values, y_values)
results = body_reg.predict(x_values)
a = float ((results [0] -results [1]) / (x [0] -x [1]))
b = float(results[1] - a * x[1])
a,b

(41075.33621453913, 2099304.086805555)

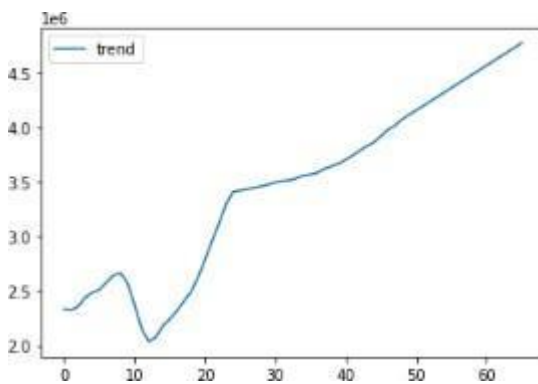
x_pred=np.array(range(48,66)).reshape(1,1)

y_pred = body_reg.predict(x_pred)

frames = [trend, pd.DataFrame(y_pred, index = x_pred.flatten(), columns=['trend'])]

trend = pd.concat(frames)
trend.plot()

```



Виведемо значення тренду.

trend

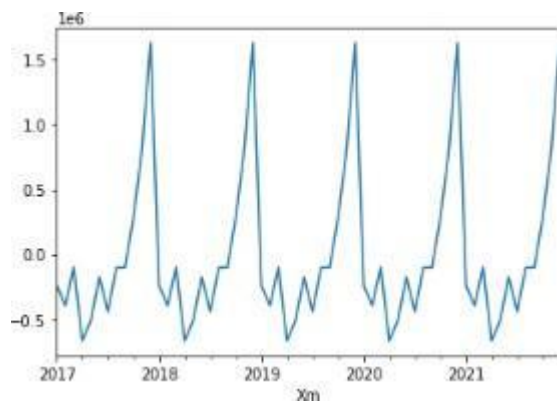
```

      trend
0  2.334994e+06
1  2.326076e+06
2  2.358670e+06
3  2.441211e+06
4  2.486578e+06
...
61 4.604900e+06
62 4.645975e+06
63 4.687050e+06
64 4.728126e+06
65 4.769201e+06
66 rows x 1 columns

```

Виведемо значення сезонності

```
seasonal_estimate.plot()
```



```
seasonal = pd.DataFrame(seasonal_estimate)
```

```
seasonal_term = seasonal.iloc[-12:]
```

```
seasonal_term
```

```
res = pd.concat([seasonal, seasonal_term], ignore_index=True)
```

```
res = res[6:]
```

```
res = res.reset_index(drop=True)
```

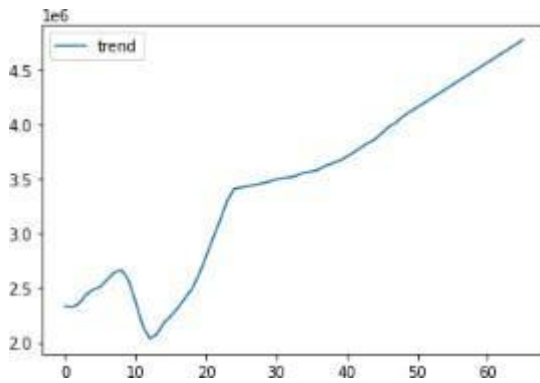
```
res
```

```

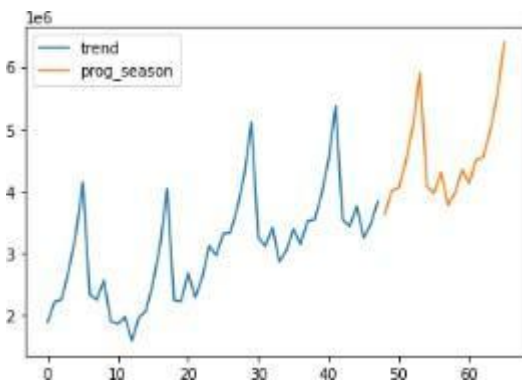
      seasonal
0  -4.382796e+05
1  -1.020162e+05
2  -1.010832e+05
3   2.855521e+05
4   8.064797e+05
...          ...
61  -1.020162e+05
62  -1.010832e+05
63   2.855521e+05
64   8.064797e+05
65  1.628663e+06
66 rows x 1 columns

```

```
trend.plot()
```



```
trend['prog_season'] = trend['trend'] + res['seasonal'].values
trend['trend'] = trend['trend'] + res['seasonal'].values
trend['prog_season'][0:48] = np.nan
trend['trend'][48:] = np.nan
trend.plot()
```



```
data.iloc[-36:]
```

Xm		Y	
2019-01-01	2599300.0	2020-07-01	3535891.0
2019-02-01	2843485.0	2020-08-01	3844218.0
2019-03-01	3328551.0	2020-09-01	3852107.0
2019-04-01	2435684.0	2020-10-01	3991024.0
2019-05-01	2909290.0	2020-11-01	4236044.0
2019-06-01	3035527.0	2020-12-01	5193643.0
2019-07-01	3206635.0	2021-01-01	3209207.0
2019-08-01	3842120.0	2021-02-01	3443479.0
2019-09-01	3721218.0	2021-03-01	3572518.0
2019-10-01	3692009.0	2021-04-01	3039082.0
2019-11-01	3925217.0	2021-05-01	3523724.0
2019-12-01	5270964.0	2021-06-01	3700062.0
2020-01-01	2749207.0	2021-07-01	4092781.0
2020-02-01	2993479.0	2021-08-01	4249313.0
2020-03-01	3477284.0	2021-09-01	4299709.0
2020-04-01	2586093.0	2021-10-01	4872711.0
2020-05-01	3100257.0	2021-11-01	4831135.0
2020-06-01	3200753.0	2021-12-01	5436279.0

Порахуємо на виведемо на екран похибки на тестових множинах

```
error = (data.iloc[-36:].values - trend['prog_season'][-48:-12].values)*(data.iloc[-36:].values - trend['prog_season'][-48:-12].values)
```

```
mse = error.mean()
```

```
mse
```

```
1277474834345.2266
```

```
rmse = np.sqrt(mse)
```

```
rmse
```

```
1130254.3228606677
```

2.5 Метод групового урахування аргументів

Метод групового урахування аргументів (МГУА) є ефективним при моделюванні складних багатопараметричних даних, де кожен параметр або функція може вносити важливий внесок у точність моделі. Використання цього методу дозволяє забезпечити більш точне відтворення даних та отримати кращі результати порівняно з використанням простіших моделей.

Цей метод базується на припущенні, що складна модель має більшу точність у відтворенні багатопараметричних даних, ніж простіші моделі. Тому метод групового урахування аргументів починає з будування простих моделей, які мають невелику кількість параметрів. Наступний крок полягає у виборі найкращої моделі серед цих простих моделей, використовуючи як критерій оцінки перевірку моделі на незалежних даних.

Після вибору найкращої простої моделі, вона розширяється додаванням нових параметрів або додаткових функцій, що дозволяють врахувати більше деталей або складніший характер вихідних даних. Потім знову робиться вибір найкращої моделі серед розширених варіантів, і процес продовжується до досягнення задовільної точності моделювання або до виконання певних умов зупинки.

Після внесення даних у програмне забезпечення GMDH Shell отримали такі результати на рис. 2.2. Бачимо, що навчання часового ряду для подальшого його прогнозування почалося 01.03.2019 та закінчилось 01.11.2021.

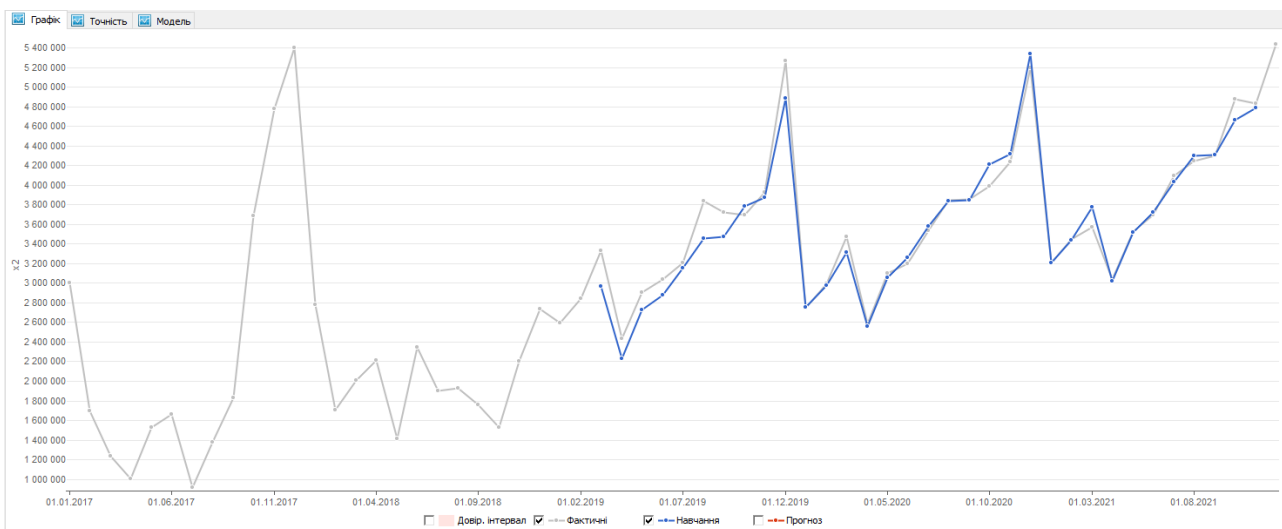


Рис. 2.2. Графік фактичних та навчальних значень

Для оцінювання точності прогнозу використаємо «екзамен» (сховані дані).

Горизонт прогнозу Екзамен

На рис. 2.3 використовуються різні параметри для прогнозування більш адекватної моделі, серед яких основним алгоритмом побудови є нейромережа МГУА.

Нейронні мережі типу GMDH, також відомі як поліноміальні нейронні мережі, використовують комбінаторний алгоритм для оптимізації з'єднання нейронів. Алгоритм ітеративно створює шари нейронів із двома або більше входами. Алгоритм зберігає лише обмежений набір оптимально складних нейронів, які ми позначаємо як початкову ширину шару.

Кожен новий шар створюється за допомогою двох або більше нейронів, взятих з будь-якого з попередніх шарів. Кожен нейрон у мережі застосовує функцію передачі (зазвичай із двома змінними), що дозволяє вичерпним комбінаторним пошуком вибрати функцію передачі, яка найбільш точно прогнозує дані тестування.

Передатна функція зазвичай має квадратичну або лінійну форму, але її також можна налаштувати в модулі розв'язувача.

Мережі типу GMDH створюють багато рівнів, але з'єднання рівнів настільки рідкісні, що їх кількість може становити лише кілька з'єднань на рівень.

Як ми вже згадували вище, алгоритм повертає лише обмежену кількість нейронів, визначених користувачем з кожного рівня; Оскільки кожен новий шар може з'єднуватися з попередніми шарами, ширина шару постійно зростає.

Беручи до уваги те, як рідко верхні шари покращують заповнення моделей, ми ділимо додатковий розмір наступного шару на два і генеруємо лише половину нейронів, згенерованих на попередньому шарі, тобто кількість нейронів N на рівні k становить $N_k=0,5 \cdot N_{k-1}$. Ця евристика робить алгоритм швидшим, але ймовірність зниження якості моделі низька.

Створення нових шарів зазвичай припиняється, коли новий шар не може

продемонструвати кращу точність тестування, ніж попередній шар. Однак ми також припиняємо створення нових шарів, якщо помилка тестування зменшилася менше ніж на 1% або якщо кількість шарів досягла певного ліміту, який ви можете визначити.

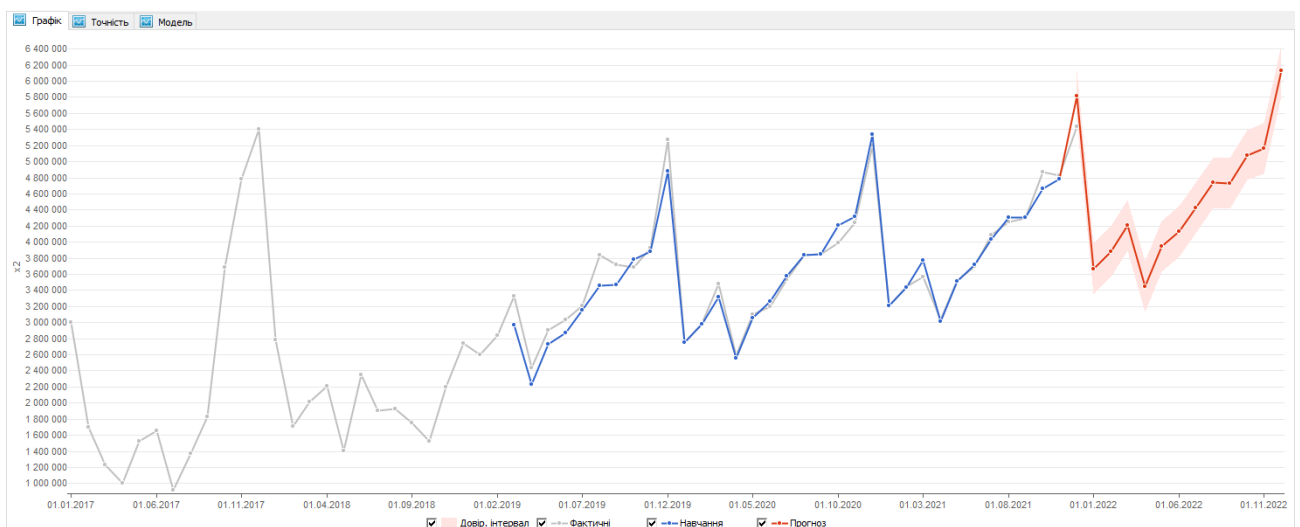
У деяких неправильно поставлених симуляціях кількість унікальних ваг мережі (або складність моделі) може бути занадто великою порівняно з кількістю рядків набору даних. Як наслідок, модель переповнює дані та може бути не в змозі передбачити належним чином вибірку. У цьому випадку ви можете обмежити кількість мережевих рівнів вручну або перепроєктувати свій набір входів, або скасувати перетворення, які надто розширюють початковий набір вхідних даних, наприклад, додаткові змінні $x_1 \cdot x_2$

The screenshot shows the MGUA software interface with the following settings:

- Перемішати спостереження: Немає
- Спосіб перевірки: ПП, виключати по одному
- Критерій: СКВ-|навчальне-перевірочн
- Ранжування змінних: Немає
- Обмежитися кращими змінними: 1
- Основний алгоритм: Нейромережа МГУА
- Функція нейрона: $a+xi+...$ (лінійн)
- Макс. кількість шарів: 24
- Початкова ширина шару: 100
- Режим часових рядів
- Задати кіль-ть потоків вручну: 7
- Заключна обробка:
 - Середнє кращих моделей: 1
 - Округлити до найближчого цілого
 - Замінити від'ємні значення нулем

Рис. 2.3. Параметри побудови моделі

Після декількох спроб підбору тих параметрів, які допоможуть побудувати найбільш адекватний і точний прогноз, маємо такі результати:



		Середня NMAE 8,44125 % Середньозважена MAPE: 6,95904 % Середня абс. сума відхилень: 6,95904 %															
№	Вихідна змінна	MAE	MAPE	Довір. інтервал	01.12.2021	01.01.2022	01.02.2022	01.03.2022	01.04.2022	01.05.2022	01.06.2022	01.07.2022	01.08.2022	01.09.2022	01.10.2022	01.11.2022	01.12.2022
1	x2	378313	6,959%	312765	5814592	3666735	3880013	4206625	3449446	3945778	4131009	4427190	4735193	4730174	5082002	5165382	6133112

2.6 Висновки до розділу 2

У розділі 2 ми спрогнозували на весь 2022 рік прибуток на прикладі магазину одягу «LC Waikiki», який саме займається роздрібними продажами. Для цього ми використали інтерактивне хмарне середовище Google Colaboratory та програмне забезпечення GMDH Shell.

Прогнозування здійснювалось за такими методами і моделями, як: модель SARIMA, модель лінійної регресії, модель Хольта-Вінтерса та метод групового урахування аргументів.

Модель SARIMA – сезонне авторегресійне інтегроване середнє ковзне, явно підтримує одновимірні дані часових рядів із сезонним компонентом. За допомогою вивчення графіка автокореляції підбрали параметри моделі та розрахували найточнішу модель.

Модель лінійної регресії відноситься до завдання визначення «лінії найкращої відповідності» через набір точок даних і стала простим попередником нелінійних методів, які використовують для навчання нейронних мереж. У ході роботи з цією моделлю вивели значення тренду та сезонності на екран.

Модель Хольта-Вінтерса враховує експоненційний тренд та адитивну сезонність, тому для прогнозу продажу товару було доцільно враховувати конкретний характер тенденції та сезонних коливань.

Основним алгоритмом побудови моделі за методом групового урахування аргументів стала нейромережа МГУА, у якої функція нейрона була лінійною.

Під час аналізу часових рядів, було пораховано середню квадратичну похибку тестових значень і виявлено, що за методом Хольта – Вінтерса ми маємо 573934550729,9542, за моделлю SARIMA – 467094455299,4019, за моделлю лінійної регресії – 1277474834345,2266, а за методом групового урахування аргументів – 143120725969. У конкретному випадку, метод

групового урахування аргументів показав найбільш точний прогноз тестових даних, оскільки має найменшу середню квадратичну похибку. Це може свідчити про те, що цей метод показує кращі результати в прогнозуванні часових рядів порівняно з іншими методами, такими як метод Хольта – Вінтерса, модель SARIMA та модель лінійної регресії. В результаті, використання методу групового урахування аргументів може бути більш ефективним для прогнозування майбутніх значень часового ряду.

ВИСНОВКИ

Одна з важливих складових успіху компанії – якісне прогнозування продажів. Правильно розрахований прогноз дозволяє більш ефективно вести бізнес, насамперед, контролювати та оптимізувати витрати. Крім того, якщо йдеться про продукцію, це дозволяє сформувати оптимальні (а не завищені чи занижені) запаси продукції на складі. Дуже важливо, щоб менеджер з продажу мав уявлення про те, що відбудеться в майбутньому, оскільки це допоможе йому планувати свої дії у разі виникнення тих чи інших подій. Багато менеджерів з продажу не визнають того, що прогнозування обсягу продажів є одним із їхніх обов'язків і залишають це на розсуд бухгалтерів, яким необхідно займатися прогнозуванням для складання бюджетів. Можливо, менеджери з продажу просто не розуміють, навіщо їм необхідне таке прогнозування, тому що вважають, що набагато важливішим їх завданням є самі продажі. І справді, завдання прогнозування менеджером із продажу формулюється часто нечітко і тому виконується так само: квапливо, без відповідної наукової бази.

У ході аналізу часових рядів, ми використовували різні методи прогнозування, такі як: метод Хольта – Вінтерса, модель SARIMA, модель лінійної регресії та метод групового урахування аргументів. Порахували середню квадратичну похибку тестових значень і виявили, що за методом Хольта – Вінтерса ми маємо 573934550729,9542, за моделлю SARIMA – 467094455299,4019, за моделлю лінійної регресії – 1277474834345,2266, а за методом групового урахування аргументів – 143120725969. Побачили, що за результатами середньої квадратичної похибки метод групового урахування аргументів має найточніший прогноз тестових даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бідюк П.І., Романенко В.Д., Тимошук О.Л. Аналіз часових рядів.: навч. посіб./ К.: НТУУ «КПІ», 2013. – 600 с.
2. Іванюта П.В. Управління ресурсами та витратами. 2-ге видання.: Центр навчальної літератури (ЦУЛ). 2019. – 320 с.
3. Йенс Нордфальт. Рітейл-маркетинг. Практики та дослідження.: видав. Альпіна Паблішер. 2018. – 512 с.
4. Керимов О. К. Аналіз та прогнозування часових рядів.: видав. РУДН, 2005. – 138 с.
5. Керимов О. К. Аналіз та прогноз фінансових часових рядів.: видав. Прометей, 2018. – 177 с.
6. «ПРОГНОЗУВАННЯ. Моделі, Методи, Алгоритми». URL: https://www.researchgate.net/publication/281110176_PROGNOZIROVANIE_Model_i_Metody_Algoritmy.
7. Aileen Nielsen Practical Time Series Analysis: Prediction with Statistics and Machine Learning.: O'Reilly Media, Inc. 2019. – 504 с.
8. «Analysis and forecasting time series for retail sales». URL: <https://bit.ly/3o8F0sJ>.
9. Ernesto Lee. Data Analytics with Python.: видав. Independently published, 2021. – 458 с.
10. Ruey S. Tsay. Multivariate Time Series Analysis.: Wiley Media, 2013. – 520 с.
11. Wes McKinney. Python та аналіз даних. 2-ге видання.: ДМК Пресс. 2020. – 540 с.
12. Madala, H. R., Ivakhnenko, A.G. Inductive learning algorithms for complex systems modeling. - New York: Boca Raton, CRC Press, 1994. - 384 p.
13. Self-organizing methods in modeling: GMDH type algorithms / Ed. S.J.Farlow. - New York, Basel: Marcel Decker Inc., 1984. - 350 p.

14. Сбірник наукових статей "Індуктивне моделювання складних систем".
- випуск 4, 2012.
15. Ivachnenko A.G., Muller J.A. Selbstorganisation von Vorhersagemodellen.
-
Berlin: Veb Verlag Technik, 1984. - 223 p.
16. Степашко В.С. Структурна ідентифікація моделей як задача відновлення сигналу в умовах неповноти інформації // Наукові праці ДНТУ. Серія: Обчисл. техніка та автоматизація. - Вип. 48. - Донецьк: ДНТУ, 2002. - 345-353 с.
17. "Моделювання та керування станом еколого-економічних систем регіону". – Випуск 4, 2008.

Додаток А

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки		
1									
2					Документація				
3									
4	САУ.КР.23.01.ПЗ				Пояснювальна записка	58	Формат А4		
5									
6					Демонстраційний матеріал	18	Презентація на CD-R		
7									
8					Копія роботи	1	Диск CD-R		
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
					САУ.КР.23.01.ДА.ПЗ.				
Змін.	Аркуш	№ докум.	Підпис	Дата					
Розроб.		Андреев І. В.			Матеріали кваліфікаційної роботи	Літ.	Аркуш	Аркушів	
К. розд.		Желдак Т. А.							
Керівн.		Желдак Т. А.				НТУ «ДП», 12; 124м-22-1			
Н.контр.		Хом'як Т. В.							
Зав. каф.		Желдак Т. А.							

Додаток Б

Відгук

на кваліфікаційну роботу магістра
студента групи 124м – 22 – 1
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: «Дослідження методів аналізу та прогнозування часових рядів роздрібних продажів».

Обсяг кваліфікаційної роботи 58 стор.

Мета кваліфікаційної роботи: розглянути теоретичні основи прогнозування часових рядів, порівняти результати залучення кількох методів прогнозування для числового ряду прибутку магазину одягу.

Тема роботи є актуальною, оскільки отримання адекватного прогнозу прибутку з урахуванням фактору сезонності є нетривіальним завданням, що виступає складовою частиною більшості задач бізнес-планування.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 124 Системний аналіз.

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня магістра. **Оригінальність наукових рішень** полягає в залученні чотирьох різних методів до прогнозування нестационарного часового ряду, причому один з них був направлений на прогнозування безпосередньо трендової складової з подальшим накладанням сезонної складової на результат прогнозу.

Практичне значення результатів кваліфікаційної роботи полягає в тому, що запропонована розроблена система допомагає контролювати та оптимізувати витрати, формувати оптимальні запаси продукції на складі.

Висновки підтверджують можливість використання результатів роботи при бізнес-плануванні роботи магазину одягу.

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано **самостійно, відповідно до завдання та у повному обсязі.**

У роботі відзначено такі недоліки:

1. У теоретичній частині описувались методи роботи з багатовимірними рядами, розробка спеціальних факторів для дослідження часових рядів, але в практичній частині розглянуто одновимірний ряд.

Кваліфікаційна робота в цілому заслуговує оцінки: **«відмінно» (90 балів).**

З урахуванням висловлених зауважень автор **заслуговує** присвоєння освітньої кваліфікації «магістр з системного аналізу».

Керівник кваліфікаційної роботи магістра,
Зав.кафедрою САУ, к.т.н., доцент

_____ / Желдак Т.А.

Додаток В**Прибуток магазину за 5 років**

Xm	Y
01.01.2017	3007542
01.02.2017	1699306
01.03.2017	1235783
01.04.2017	1005879
01.05.2017	1529041
01.06.2017	1660054
01.07.2017	916857
01.08.2017	1376223
01.09.2017	1833750
01.10.2017	3685075
01.11.2017	4783007
01.12.2017	5398573
01.01.2018	2785230
01.02.2018	1707580
01.03.2018	2009750
01.04.2018	2212900
01.05.2018	1410825
01.06.2018	2348528
01.07.2018	1905324
01.08.2018	1928276
01.09.2018	1757532
01.10.2018	1525121
01.11.2018	2200500
01.12.2018	2738434
01.01.2019	2599300
01.02.2019	2843485
01.03.2019	3328551
01.04.2019	2435684
01.05.2019	2909290
01.06.2019	3035527
01.07.2019	3206635
01.08.2019	3842120
01.09.2019	3721218
01.10.2019	3692009
01.11.2019	3925217
01.12.2019	5270964
01.01.2020	2749207
01.02.2020	2993479
01.03.2020	3477284
01.04.2020	2586093
01.05.2020	3100257

01.06.2020	3200753
01.07.2020	3535891
01.08.2020	3844218
01.09.2020	3852107
01.10.2020	3991024
01.11.2020	4236044
01.12.2020	5193643
01.01.2021	3209207
01.02.2021	3443479
01.03.2021	3572518
01.04.2021	3039082
01.05.2021	3523724
01.06.2021	3700062
01.07.2021	4092781
01.08.2021	4249313
01.09.2021	4299709
01.10.2021	4872711
01.11.2021	4831135
01.12.2021	5436279