

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

Кафедра технологій управління

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: «Інформаційна аналітика та впливи»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему:

**“Інформаційне забезпечення прийняття управлінських рішень  
методами аналізу часових рядів”**

**Студента 2-го курсу групи ІАВ-21**

Дейнеги Владислава Олександровича  
(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 – 2023**

**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.  
« \_\_\_ » \_\_\_\_\_ 20\_\_ року

**ЗАВДАННЯ  
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент Дейнега Владислав Олександрович  
Група ІАВ-21

- 1. Тема кваліфікаційної роботи:** “Інформаційне забезпечення прийняття управлінських рішень методами аналізу часових рядів”  
Затверджена наказом по від « 08 » грудня 2022 р. № 5 .
- 2. Строк подання студентом готової роботи** – “16” травня 2023 р.
- 3. Цільова установка та вихідні дані до роботи:** дослідження застосування часових рядів у різних сферах діяльності, розробка автоматизованої системи прогнозування часових рядів, яка дає змогу аналізувати та прогнозувати майбутні значення часових рядів на основі методів прогнозування
- 4. Зміст роботи:** вивчення теоретичних засад аналізу часових рядів, включно з методами та підходами до аналізу, попередньою обробкою даних, побудовою прогнозів тощо, дослідження практичного застосування прогнозування часових рядів у різних галузях і сферах діяльності, вивчення інструментів і технологій для розроблення інформаційного забезпечення, які дадуть змогу створити систему автоматизованого прогнозування часових рядів, розроблення системи автоматизованого прогнозування часових рядів, що включає попередню обробку даних, вибір методів прогнозування та побудова прогнозів, демонстрація функціоналу системи на тестових даних для перевірки працездатність і ефективність розробленої системи
- 5. Перелік графічного матеріалу (слайдів)** 20 рисуноків, 10 формул, 1 додаток, 28 слайдів презентації доповіді

**6. Календарний план виконання роботи:**

№ п/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1.	Вибір теми дипломної роботи	3	01.10.22	01.10.22
2.	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	27.12.22	27.12.22
3.	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	08.01.23	08.01.23
4.	Складання розгорнутого плану кваліфікаційної роботи	5	18.01.23	18.01.23
5.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	5	19.01.23 - 20.01.23	20.11.23
6.	Підготовка розділу 1 «Теоретичні основи прогнозування часових рядів»	10	12.02.23	12.02.23
7.	Підготовка розділу 2 «Формалізація моделей, методів та технологій прогнозування часових рядів для прийняття управлінських рішень»	14	08.03.23	08.03.23
8.	Підготовка розділу 3 «Моделювання системи автоматизованого прогнозування часових рядів»	14	01.04.23	01.04.23
9.	Підготовка розділу 4 «Демонстрація функціоналу системи прогнозування методами часових рядів»	13	20.04.23	20.04.23
10.	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	15	03.05.23	03.05.23
11.	Передача кваліфікаційної роботи науковому керівникові	2	04.05.23	04.05.23
12.	Передача кваліфікаційної роботи рецензенту для рецензування	2	11.05.23	11.05.23
13.	Попередній захист кваліфікаційної роботи	5	17.05.23	17.05.23

Дата видачі завдання « 08 » грудня 2022 р.

Керівник роботи доктор технічних наук, доцент Хлевна Юлія Леонідівна  
(посада, прізвище, ім'я, по батькові)

---

(підпис)

Завдання прийняв до виконання студент групи ІАВ-21

Дейнега Владислав Олександрович  
(прізвище, ім'я, по батькові)

---

(підпис)

## ЗМІСТ

<b>АНОТАЦІЯ.....</b>	<b>7</b>
<b>ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ.....</b>	<b>8</b>
<b>ВСТУП.....</b>	<b>9</b>
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....</b>	<b>13</b>
1.1 Вступ до розділу.....	13
1.2 Аналіз практичного застосування часових рядів при прийнятті управлінських рішень.....	14
1.3 Аналіз часових рядів у бізнесі.....	16
1.4 Використання методів ARIMA та експоненціального згладжування для прогнозування часових рядів.....	18
1.5 Попередня оброблення часових рядів.....	24
1.6 Виявлення аномалій у часових рядах.....	26
1.7 Аналіз гібридних моделей.....	29
1.8 Автоматизація часових рядів.....	31
1.9 Система підтримки ухвалення рішень із прогнозуванням.....	33
1.10 Висновок по першому розділу.....	34
<b>РОЗДІЛ 2. ФОРМАЛІЗАЦІЯ МОДЕЛЕЙ, МЕТОДІВ ТА ТЕХНОЛОГІЙ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ДЛЯ ПРИЙНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ.....</b>	<b>36</b>
2.1 Особливості моделей часових рядів при прогнозуванні.....	36
2.2 Моделювання часових рядів.....	40
2.2.1 Просте експоненціальне згладжування.....	40
2.2.2 Подвійне експоненціальне згладжування.....	41
2.2.3 Модель Хольта-Вінтерса.....	42
2.2.4 AR модель.....	44
2.2.5 MA модель.....	45
2.2.6 ARMA модель.....	47
2.2.7 ARIMA модель.....	48
2.2.8 SARIMA модель.....	49
2.3 Параметри моделі ARIMA.....	50
2.3.1 ACF та PACF.....	51
2.3.2 Автоматичний підхід пошуку параметрів у сімействі моделі ARIMA.....	52
2.3.3 Формалізація підходів до аналізу параметрів у сімействі моделі ARIMA.....	53
2.4 Система автоматизованого прогнозування часових рядів.....	53

2.5 Концепція інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів.....	54
2.5.1 Інструменти для інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів.....	56
2.5.2 Концептуальна модель інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів.....	59
2.6 Висновок по другому розділу.....	64
<b>РОЗДІЛ 3. МОДЕЛЮВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....</b>	<b>66</b>
3.1 Вступ до розділу.....	66
3.2 Система автоматизованого прогнозування часових рядів.....	66
3.3 Висновок по третьому розділу.....	91
<b>РОЗДІЛ 4. ДЕМОНСТРАЦІЯ ФУНКЦІОНАЛУ СИСТЕМИ ПРОГНОЗУВАННЯ МЕТОДАМИ ЧАСОВИХ РЯДІВ.....</b>	<b>94</b>
4.1 Вступ до розділу.....	94
4.2 Використання системи.....	94
4.3 Висновок по четвертому розділу.....	105
<b>ВИСНОВКИ.....</b>	<b>107</b>
<b>СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....</b>	<b>112</b>
<b>ДОДАТКИ.....</b>	<b>117</b>

## **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 Дейнеги Владислава Олександровича.

Тема роботи – «Інформаційне забезпечення прийняття управлінських рішень методами аналізу часових рядів».

Мета дипломної роботи магістра – автоматизація прийняття управлінських рішень за рахунок розробки інформаційного забезпечення, в основі якого методи аналізу та прогнозування (часові ряди). У межах цієї мети створити систему автоматизованого прогнозування часових рядів, що дає змогу користувачам отримувати прогнозні значення часових рядів за допомогою кількох методів і мінімального залучення до процесу аналізу даних.

Об'єкт дослідження – є процеси інформаційного забезпечення прогнозування часових рядів.

Предмет дослідження – є моделі, методи та інформаційні та програмні засоби прогнозування часових рядів.

Наукова новизна роботи – розвиток моделі прогнозування часових рядів, в основі якої аналіз результатів декількох моделей аналізу даних із мінімальним залученням користувача до процесу аналізу. Така модель буде допомагати в прийнятті раціональних управлінських рішень за рахунок розробки інформаційного забезпечення.

Практичне значення роботи - розроблено інформаційне забезпечення прийняття управлінських рішень методами аналізу часових рядів та систему автоматизованого прогнозування часових рядів.

Дипломна робота складається зі вступу, основної частини, яка включає чотири розділи, висновків та списку використаних джерел. Всього налічує 126 сторінок та перелік посилань з 43 джерел на 5 сторінках.

Ключові слова: часові ряди, прогнозування, аналіз даних, інформаційне забезпечення, автоматизована система.

## **ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ**

ARIMA - Autoregressive Integrated Moving Average

AIC - Akaike Information Criterion

MAPE - Mean Absolute Percentage Error

STL - Seasonal and Trend decomposition using Loess

ETS - Error, Trend, Seasonal

ACF - Autocorrelation Function

PACF - Partial Autocorrelation Function

## ВСТУП

### Актуальність дослідження

Аналіз і прогнозування часових рядів - це важлива галузь, яка застосовується в різних галузях і сферах, таких як фінанси, економіка, маркетинг, метеорологія та інші. Ця галузь вивчає зміни в часі як для одного показника, так і для кількох показників, і дає змогу зрозуміти, як ці зміни пов'язані між собою та які можуть бути майбутні тенденції.

Актуальність аналізу та прогнозування часових рядів впливає з того, що нині дані збирають і записують із високою швидкістю, і це дає нам змогу використовувати ці дані для створення точних і достовірних прогнозів. Завдяки розвитку технологій і програмного забезпечення, аналіз часових рядів став більш доступним і простим у використанні.

Нині прогнозування часових рядів відіграє важливу роль у бізнесі. Це дає змогу компаніям ухвалювати правильні рішення на основі даних, що, своєю чергою, може призвести до поліпшення продуктивності, скорочення витрат і підвищення прибутку. У фінансовій сфері аналіз часових рядів допомагає інвесторам ухвалювати правильні рішення і управляти своїми портфелями. Особливо це зручно реалізовувати, якщо бізнес забезпечений автоматизованими системами прогнозування.

Аналіз і прогнозування часових рядів також має важливе значення в науці та дослідженнях. Це допомагає вченим досліджувати зміни в різних галузях, як-от клімат, охорона здоров'я, економіка, і багато іншого.

Так як, аналіз і прогнозування часових рядів дають змогу точніше розуміти минулі тенденції та передбачати майбутні зміни, що є важливим інструментом для державних і муніципальних організацій під час планування інфраструктури та громадських послуг.

Дослідженням часових рядів займалось багато вчених: Джордж Е. П. Бокс та Гвілім Дженкінс "Time Series Analysis: Forecasting and Control"[1], Роберт Ф. Інгл "Autoregressive Conditional Heteroskedasticity with Estimates of the

Variance of United Kingdom Inflation"[2], Джеймс Д. Гамільтон "Time Series Analysis"[3], Роберт Хайндман та Джордж Атанасопулос "Forecasting: Principles and Practice"[4], Питер Броквелл та Річард А. Девіс "Introduction to Time Series and Forecasting"[5], Теренс К. Міллс "The Foundations of Modern Time Series Analysis"[6] та інші.

Одним із ключових аспектів аналізу та прогнозування часових рядів є використання математичних моделей, таких як авторегресійні моделі, ковзні середні та інші. Ці моделі допомагають передбачати майбутні значення на основі минулих даних і розуміти, які фактори можуть вплинути на ці значення.

Однак, необхідно розуміти, що аналіз, прогнозування часових рядів мають свої обмеження і вони можуть бути недостатньо точними в деяких випадках. Крім того, Наприклад, якщо відбуваються несподівані події, які не враховуються в моделі, це може призвести до неточних прогнозів.

Питання інформаційного забезпечення прогнозування відображено у таких авторів як Роберт Філдс, Спірос Макридакіс, Моніка Адя та інші.

Встановлено, що питання інформаційного забезпечення прогнозування часових рядів потребує розвитку та подальшого дослідження.

Розвиток моделей, методів, а також інформаційного забезпечення прогнозування часових рядів є актуальною задачею.

**Мета роботи** - автоматизація прийняття управлінських рішень за рахунок розробки інформаційного забезпечення, в основі якого методи аналізу та прогнозування (часові ряди). У межах цієї мети створити систему автоматизованого прогнозування часових рядів, що дає змогу користувачам отримувати прогнозні значення часових рядів за допомогою кількох методів і мінімального залучення до процесу аналізу даних.

Для досягнення мети розроблення інформаційного забезпечення для прийняття управлінських рішень на основі аналізу часових рядів необхідно було вирішити такі **завдання**:

- Вивчення теоретичних засад аналізу часових рядів, включно з методами та підходами до аналізу, попередньою обробкою даних, побудовою прогнозів тощо.
- Дослідження практичного застосування прогнозування часових рядів у різних галузях і сферах діяльності.
- Вивчення інструментів і технологій для розроблення інформаційного забезпечення, які дадуть змогу створити систему автоматизованого прогнозування часових рядів для прийняття управлінських рішень.
- Розроблення системи автоматизованого прогнозування часових рядів, що включає попередню обробку даних, вибір методів прогнозування та побудова прогнозів.
- Демонстрація функціоналу системи на тестових даних для перевірки працездатності і ефективності розробленої системи при прийнятті управлінських рішень.

**Об'єкт дослідження** – є процеси інформаційного забезпечення прогнозування часових рядів.

**Предмет дослідження** – є моделі, методи та інформаційні та програмні засоби прогнозування часових рядів.

**Наукова новизна роботи** – розвиток моделі прогнозування часових рядів, в основі якої аналіз результатів декількох моделей аналізу даних із мінімальним залученням користувача до процесу аналізу. Така модель буде допомагати в прийнятті раціональних управлінських рішень за рахунок розробки інформаційного забезпечення.

Практичне значення роботи - розроблено інформаційне забезпечення прийняття управлінських рішень методами аналізу часових рядів та систему автоматизованого прогнозування часових рядів.

**Методи дослідження.** Теоретичною основою дослідження стали загальнонаукові методи пізнання: аналітичний, індуктивний, дедуктивний, системний. Математичний апарат для прогнозування включає різні методи

аналізу часового ряду такі як STL декомпозицію, Exponential smoothing та Auto ARIMA. Для вирішення завдання розробки інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів були використані такі інструменти, як мова програмування Python, Jupyter Notebook, PyCharm.

**Публікації:**

Deineha V., Mochalova D., ANALYSIS OF FORECASTING METHODS FOR FINANCIAL MARKETS, Information Technology and Implementation (IT&Is-2021)

Deineha V., Khlevna I., PREDICTING FRAUDULENT CRYPTOCURRENCY TRANSACTIONS USING LOGISTIC REGRESSION, International Journal of Information and Communication Technologies

**Апробація роботи:** Автор виступав доповідачем на VIII International conference “Information Technology and Implementation” (IT&Is-2021).

**Структура роботи:** Кваліфікаційна робота складається зі вступу, 4 розділів, висновків, списку літератури з 43 пунктів та 1 додаток. Загальний обсяг кваліфікаційної роботи становить 126 сторінок, із них 111 сторінок основного тексту, який містить 20 рисунків.

# РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

## 1.1 Вступ до розділу

У цьому розділі було проведено аналіз літератури, пов'язаний із темою магістерського дослідження, щоб розглянути різні методи, які використовуються в практичній частині дослідження. Було досліджено різні джерела, включно з науковими статтями та іншими публікаціями, щоб визначити, які методи та інструменти вже використовуються в цій галузі та яких результатів було досягнуто.

Було зосереджено увага на аналізі різних методів, використовуваних у попередніх дослідженнях, а також визначись їхні переваги та недоліки. Також було звернуто увагу на нові методи, які можуть бути застосовані в даній галузі, і описуємо їхні особливості.

У процесі аналізу було звернуто увагу на різні аспекти, такі як методи статистичного аналізу, автоматизація, аналіз часових рядів та інші. Також були розглянуті різні підходи до моделювання даних, включно з методами прогнозування та попередньої обробки даних.

Основну увагу в огляді літератури було приділено таким розділам:

- Методи прогнозування часових рядів. Було розглянуто методи прогнозування часових рядів, такі як методи експоненціального згладжування, методи ARIMA та інші.
- Застосування методів аналізу та прогнозування часових рядів у різних галузях. Було розглянуто приклади застосування методів аналізу та прогнозування часових рядів у різних галузях, таких як економіка, фінанси, медицина, екологія та інші.
- Попереднє оброблення часових рядів. Було розглянуто методи попередньої обробки часових рядів, такі як обробка пропущених значень.

- Виявлення аномалій у часових рядах. Було розглянуто методи виявлення аномалій у часовому ряді для виявлення та обробки аномальних точок даних.
- Гібридні моделі. Було розглянуто гібридні моделі прогнозування часових рядів, які об'єднують різні методи та підходи.
- Автоматизація часових рядів. Було розглянуто наявні інструменти та методи автоматизації процесу аналізу та прогнозування часових рядів, такі як автоматичні методи вибору моделей, автоматичний підбір гіперпараметрів та інші.
- Система підтримки ухвалення рішень із прогнозуванням. Було розглянуто систему підтримки ухвалення рішень із прогнозуванням, що надає організаціям та окремим особам інструменти й аналітичні можливості для ухвалення більш ґрунтовних та поінформованих рішень, що ґрунтуються на прогнозах і передбаченнях майбутніх подій і тенденцій.

Мета аналізу літератури - провести аналіз існуючих методів, та які методи є більш ефективними в певних випадках аналізу часових рядів, які можна застосовувати в нашому магістерському дослідженні.

## **1.2 Аналіз практичного застосування часових рядів при прийнятті управлінських рішень.**

Аналіз часових рядів є важливим інструментом для прийняття управлінських рішень у різних сферах, таких як фінанси, економіка, маркетинг, виробництво, енергетика тощо. Актуальність аналізу часових рядів зумовлена тим, що багато явищ і процесів, які підлягають дослідженню, змінюються в часі та мають тенденції, цикли, сезонність і випадкову складову.

Аналіз часових рядів дає змогу виявити та проаналізувати ці характеристики, що допомагає зрозуміти динаміку явища чи процесу,

прогнозувати його поведінку в майбутньому та ухвалювати управлінські рішення на основі отриманих даних.

Наприклад, аналіз часових рядів фінансових показників дає змогу оцінити поточний стан фірми, виявити тенденції та прогнозувати її майбутній фінансовий стан. Це може бути корисно для ухвалення рішень про розширення бізнесу, інвестування, управління ризиками тощо.

Аналіз часових рядів у маркетингу дає змогу оцінити ефективність рекламних кампаній, виявити тенденції в поведінці споживачів і прогнозувати попит на товари та послуги. Це може бути корисно для розроблення маркетингових стратегій, управління запасами тощо.

Аналіз часових рядів у виробництві дає змогу оцінити ефективність виробничих процесів, виявити тенденції у зміні обсягів виробництва і прогнозувати попит на продукцію. Це може бути корисно для оптимізації виробничих процесів, управління запасами тощо.

Аналіз часових рядів в енергетиці дає змогу оцінити ефективність використання ресурсів, виявити тенденції у зміні споживання енергії та прогнозувати попит на енергію. Це може бути корисно для оптимізації використання енергоресурсів, планування інвестицій тощо.

Крім того, аналіз часових рядів може допомогти у виявленні аномалій, викидів та інших їхніх нетипових значень, що можуть свідчити про проблеми або несподівані події. Наприклад, у фінансовій сфері аномалії можуть бути пов'язані з фінансовим шахрайством, а у виробничій сфері - з відмовами обладнання або іншими технічними проблемами.

Шляхом аналізу часових рядів можна виявити такі аномалії, а також спробувати нівелювати їхній вплив на прогноз у майбутнє, якщо є така потреба. Це дає змогу швидко реагувати на проблеми і вживати заходів щодо їх усунення або мінімізації.

Крім того, аналіз часових рядів може використовуватися для оцінки ефективності прийнятих управлінських рішень. Наприклад, якщо було внесено зміни у виробничі процеси, можна проаналізувати зміни в часових рядах

виробництва, подивитися, наскільки змінилися прогнозні значення та оцінити, наскільки ці зміни були ефективними.

Таким чином, аналіз часових рядів є необхідним інструментом для прийняття управлінських рішень у різних сферах. Він дає змогу виявляти тенденції, прогнозувати поведінку явищ і процесів, виявляти аномалії та оцінювати ефективність ухвалених рішень. Тому, розуміння методів аналізу часових рядів та вміння застосовувати їх на практиці, може допомогти в ухваленні правильних рішень та досягненні успіху в бізнесі.

### **1.3 Аналіз часових рядів у бізнесі**

Часові ряди - це послідовності даних, зібраних у різні моменти часу. У бізнесі аналіз часових рядів є важливим інструментом для ухвалення управлінських рішень і визначення тенденцій у виробництві, продажах, фінансах та інших галузях.

Однією з основних причин використання аналізу часових рядів є можливість прогнозування майбутніх значень на основі минулих даних. Це може допомогти бізнесу визначити попит на продукти і послуги, прогнозувати доходи і витрати, планувати виробничий обсяг і багато іншого.

Аналіз часових рядів також може допомогти виявити циклічність, тренди і сезонність даних, що може допомогти бізнесу визначити, як краще розподілити ресурси і управляти змінами.

Застосування методів аналізу часових рядів може бути корисним для різних галузей бізнесу, таких як виробництво, фінанси, маркетинг і логістика. Наприклад, аналіз часових рядів може допомогти виробникам визначити, яку кількість продукції слід виробляти в певний період часу, щоб задовольнити потреби споживачів. Для фінансових установ аналіз часових рядів може бути корисним під час прогнозування доходів і витрат, оцінки ризиків і визначення інвестиційних можливостей.

Крім того, аналіз часових рядів може бути корисним для маркетингових і рекламних компаній, щоб визначити ефективність рекламних кампаній, що проводяться в певну пору року або в певних регіонах.

Загалом, аналіз часових рядів є важливим інструментом для ухвалення управлінських рішень і визначення тенденцій у різних галузях бізнесу.

Аналіз часових рядів у бізнесі стає дедалі актуальнішим нині. З появою великої кількості даних, що надаються різними системами, стало можливим отримувати точніші та своєчасніші прогнози, що зі свого боку допомагає підприємствам ухвалювати більш обґрунтовані управлінські рішення.

Однією з основних переваг аналізу часових рядів є можливість виявлення та аналізу різноманітних трендів і циклічних коливань, які можуть впливати на результати бізнесу. Це дає змогу підприємствам точніше прогнозувати попит на свої товари та послуги, управляти запасами та виробничими потужностями, а також оптимізувати маркетингові стратегії та покращувати якість обслуговування.

Крім того, аналіз часових рядів допомагає запобігти можливим фінансовим втратам, пов'язаним з неефективним управлінням ресурсами та неправильними управлінськими рішеннями. Також він дає змогу підприємствам швидко реагувати на зміни в економічному середовищі та адаптуватися до них.

Сучасні технології, як-от машинне навчання та штучний інтелект, дають змогу обробляти й аналізувати великі обсяги даних, що робить аналіз часових рядів ефективнішим і точнішим. Крім того, наявність широкого спектра інструментів і програмних засобів для аналізу часових рядів робить цей процес доступнішим і зручнішим для підприємств різних галузей.

#### **1.4 Використання методів ARIMA та експоненціального згладжування для прогнозування часових рядів**

Стаття [7] описує статистичний метод аналізу часових рядів, відомий як ARIMA, який використовується для прогнозування майбутніх значень часового ряду на основі його історичних даних.

Моделі ARIMA використовуються в багатьох галузях для прогнозування майбутніх значень часових рядів. Їх найширше використовують у прогнозуванні попиту, де менеджерам надають надійні рекомендації щодо ухвалення рішень у ланцюжках поставок. Крім того, моделі ARIMA можуть використовуватися для прогнозування майбутніх цін на акції на основі минулих цін.

Проте ARIMA має деякі недоліки, які можуть обмежити його ефективність у деяких випадках. Наприклад, вибір оптимальних значень параметрів  $p$  і  $q$ , може залежати від досвіду і майстерності розробника моделі, що може бути суб'єктивним. Очевидно, що тут ідеться про використання розглянутого інструменту прогнозування ручним способом налаштування моделі. Крім того, ARIMA моделі можуть бути складними і мати низьку пояснювальну здатність порівняно з більш простими моделями, такими як експоненціальне згладжування.

Висновки статті полягають у тому, що ARIMA моделі можуть бути корисними інструментами для короткострокового прогнозування, як і решта моделей, на основі даних часових рядів, але для досягнення найкращих результатів, розробники моделей мають бути обережними під час вибору оптимальних значень параметрів та використання простіших моделей, якщо вони демонструють вищу пояснювальну здатність.

Таким чином, стаття наголошує на важливості адекватного вибору моделі для прогнозування часових рядів, а також на необхідності врахування історичних даних і досвіду в розробленні моделей для досягнення найкращих результатів прогнозування.

Стаття [8] обговорює застосування моделей часових рядів для прогнозування соціально-економічних явищ і ризиків у фінансовій сфері.

Зокрема, автори представляють моделі AR, MA та ARIMA, що об'єднує обидві моделі, та застосовують їх для прогнозування ризиків на Національній фондовій торгівлі NTB. Результати аналізу відповідають реальній ситуації та допомагають у прогнозуванні фінансових ризиків.

Стаття описує метод аналізу часових рядів для прогнозування майбутніх коливань цін на акції. Історична інформація про ціни може використовуватися для оцінки ймовірності розподілу майбутніх коливань цін на акції, що дає змогу уникнути ризику величезних втрат. Зокрема виконувалася обробка різниць першого порядку для ряду цін на акції, щоб отримати стійку диференціальну послідовність, отримувалася інформація щодо стаціонарності ряду тощо.

Виходячи з цього, стаття підкреслює важливість застосування моделей часових рядів у фінансовій сфері для прогнозування ризиків і прийняття обґрунтованих рішень. Застосування ARIMA для прогнозування ризиків на NTB є практичним прикладом ефективного використання моделей часових рядів у реальному житті. Метод аналізу часових рядів, запропонований авторами, може бути корисним для прогнозування екстремальних ситуацій і допомогти знизити фінансові ризики.

Однак під час використання цих моделей і методів необхідно враховувати, що минула поведінка часових рядів не завжди може бути використана для прогнозування майбутніх значень з високою точністю. Тому, важливо проводити регулярну перевірку та аналіз результатів прогнозування, а також враховувати додаткові фактори, які можуть вплинути на майбутні тенденції та коливання цін на акції.

У цій статті [9] було проведено дослідження на тему застосування математичних моделей для прогнозування цін на акції компаній, зареєстрованих на Фондовій біржі Коломбо. Для цього було порівняно дві моделі - модель часового ряду ARIMA і модель цін акцій з використанням FBMAR.

Дослідження було проведено за період із січня 2015 року до червня 2019 року. Для перевірки стаціонарності набору даних було використано тест

Дікі-Фуллера. Тест Дікі-Фуллера є статистичним тестом, який дає змогу визначити, чи є часовий ряд стаціонарним, чи ні.

Стаціонарність часового ряду важлива для прогнозування майбутніх значень, тому що стаціонарні часові ряди мають постійне середнє і дисперсію, що спрощує прогнозування. Тест Дікі-Фуллера перевіряє нульову гіпотезу про те, що часовий ряд має одиничний корінь, що означає, що він нестационарний. Якщо значення тесту менше за рівень значущості, то нульова гіпотеза відкидається, що свідчить про те, що ряд є стаціонарним. У даній статті тест Дікі-Фуллера використовувався для перевірки стаціонарності набору вище описаних даних. Перевірка стаціонарності часових рядів є важливим кроком при побудові моделей прогнозування цін на акції.

На першому етапі дослідження було знайдено найбільш підходящу модель ARIMA за допомогою інформаційних критеріїв Акаїке, середньоквадратичної помилки і середньої абсолютної помилки у відсотках.

Критерій Акаїке - це статистичний критерій, що використовується для порівняння різних статистичних моделей на основі їхньої відповідності даним і кількості параметрів моделі.

У цьому дослідженні критерій Акаїке було використано для вибору найбільш підходящої моделі ARIMA з кількох запропонованих моделей. Зокрема, розглядали різні комбінації параметрів, таких як авторегресійний порядок (p), ступінь диференціювання (d) і порядок ковзного середнього (q) у моделі ARIMA.

Критерій Акаїке обчислює відносну ефективність кожної моделі, враховуючи її точність і складність (кількість параметрів), і обирає модель із найменшим значенням AIC. Нижче значення AIC вказує на кращу модель, яка має точніші прогнози та простішу структуру.

Використання критерію Акаїке для вибору найбільш підходящої моделі ARIMA є загальноприйнятою методологією в аналізі часових рядів. Таким чином, у даному дослідженні використання критерію Акаїке допомогло

дослідникам вибрати найбільш підходящу модель ARIMA для прогнозування цін на акції.

Крім того, було проведено тест ARCH для визначення динамічної поведінки умовної дисперсії в наборі даних. Однак, за результатами тесту, було встановлено, що ефект ARCH не присутній у даному дослідженні часового ряду.

На другому етапі дослідження було застосовано модель FBMAR для прогнозування майбутніх цін на акції. Змодельовані ціни, отримані за допомогою обох моделей, було порівняно з емпіричними цінами. Результати дослідження показали, що модель ARIMA і модель FBMAR мають потенціал для прогнозування індексів цін на акції на короткостроковій основі. Однак, модель FBMAR виявилася більш придатною для прогнозування щоденних цін закриття, ніж модель ARIMA.

Загалом, дослідження показало, що використання статистичних моделей для прогнозування цін на акції може бути ефективним інструментом у прогнозуванні. Результати дослідження показують, що обидві моделі можуть бути використані для прогнозування індексів цін на акції на короткостроковій основі. Крім того, модель FBMAR виявилася більш придатною для прогнозування щоденних цін закриття, ніж модель ARIMA.

Ця стаття [10] описує дослідження, проведене для прогнозування цін закриття кількох криптовалют, зокрема Bitcoin, Ripple, Dash, Litecoin і Ethereum. Автори використовували дані про ціни, ринковий капітал і обсяги за попередні дні та застосували статистичні методи й алгоритми машинного навчання, включно з простою і множинною лінійною регресією та моделлю довгострокової короткострокової пам'яті (LSTM), для прогнозування цін на основі історичних даних.

У результаті дослідження автори порівняли точність різних моделей прогнозування з використанням показників MAPE і відносної RMSE. Виявилось, що точність моделі ARIMA була найвищою на цьому наборі даних,

за якою слідували багатовимірною лінійною регресією і LSTM. Модель простої лінійної регресії виявилася найменш точною з усіх розглянутих моделей.

Загалом, стаття показала, що прогнозування цін на криптовалюти може бути досягнуто з використанням відповідних методів прогнозування, і що використання історичних даних може бути ефективним інструментом для аналітиків та інвесторів.

Стаття [11] описує метод прогнозування попиту на основі експоненціального згладжування. Цей метод використовує історичні дані, щоб скласти прогноз на майбутній період. Для експоненціального згладжування необхідно визначити константу згладжування, яка знаходиться в діапазоні від 0 до 1. Константа використовується як експонента у формулі прогнозу.

Однією з переваг експоненціального згладжування є його простота в розумінні та застосуванні. Для цього методу необхідний тільки останній прогноз, фактичні дані та константа згладжування. Експоненціальне згладжування також краще враховує свіжі дані, ніж старі дані, що дає змогу точніше прогнозувати майбутній попит. Крім того, прогнози, отримані за допомогою експоненціального згладжування, точні та надійні, що допомагає планувати запаси.

Однак у експоненціального згладжування є й недоліки. Наприклад, цей метод не враховує високі та низькі коливання даних, оскільки графік прогнозу показує плавну лінію даних. Крім того, цей метод найкраще підходить для короткострокового прогнозування, наприклад, на наступний період.

Таким чином, експоненціальне згладжування є корисним методом прогнозування попиту, який простий у використанні і дає точні прогнози на короткі періоди часу. Однак, для прогнозування довгострокових тенденцій, як і для будь-якого методу прогнозування, необхідно враховувати всі фактори, що можуть вплинути на попит, і використовувати кілька методів прогнозування для отримання більш точних результатів.

З цієї статті [12] випливає, що використання моделі експоненціального згладжування в прогнозуванні ринку морських перевезень нафти є ефективним

підходом. Автор доопрацював наявну модель і досяг більш точних прогнозів середнього тайм-чартерного еквівалента танкера за різними маршрутами транспортування нафти у Світовому океані в кризовий період 2015-2019 рр. Згідно з результатами, точність запропонованого методу перевершує найвні методи, методи авторегресії та моделі машинного навчання за всіма використовуваними метриками помилок. Таким чином, цей метод може бути використаний для комерційних цілей операторами і фрахтувальниками танкерного флоту з точністю в 71% випадків.

У цьому дослідженні [13] використовували методи експоненціального згладжування Брауна та ARIMA для прогнозування статистичних даних щодо COVID-19 у Туреччині. Дослідники обирали модель із найменшими значеннями критерію Акаїке для кожного показника та визначили ARIMA як найкращу модель для прогнозування шуканої кількості пов'язаної з пандемією у вищевказаній країні. Прогнозні значення стабільні в часі, і моделі ARIMA можуть використовуватися для забезпечення здоров'я і безпеки в разі свіжих спалахів епідемії. Таким чином, цей метод аналізу даних показав достатню застосовність не тільки в галузі економіки та фінансів, а й у галузі охорони здоров'я, що свідчить про його універсальність і широкі можливості застосування в різних сферах діяльності.

Згідно зі статтею [14], поширена хибна думка про те, що ARIMA-моделі є більш загальними, ніж моделі експоненціального згладжування, є помилковою. Насправді, обидва класи моделей перетинаються і доповнюють одна одну, і кожен з них має свої сильні та слабкі сторони. Лінійні моделі експоненціального згладжування є окремими випадками моделей ARIMA, але нелінійні моделі експоненціального згладжування не мають еквівалентного аналога в ARIMA.

У статті йдеться про те, що всі моделі простору станів з експоненціальним згладжуванням нестационарні, тому, якщо потрібна стаціонарна модель, краще використовувати моделі ARIMA. Однак, однією з переваг експоненціальних згладжувальних моделей є те, що вони можуть бути

нелінійними та краще моделювати часові ряди, що демонструють нелінійні характеристики, включно з гетероскедастичністю.

Висновки зі статті: моделі ARIMA та експоненціального згладжування не є взаємовиключними, і кожна з них має свої переваги. Необхідно обирати відповідну модель залежно від конкретних умов завдання, вимог до точності та особливостей даних.

### **1.5 Попередня оброблення часових рядів**

Стаття [15] описує проблеми, пов'язані з наявністю пропущених значень і викидів у часових рядах, які можуть вплинути на аналіз даних і ускладнити проведення статистичного аналізу. У статті пропонується кілька методів визначення оцінок для заміни відсутніх значень і викидів, включно з використанням середнього значення, медіани, статистики найбільшого порядку і прогнозних значень, заснованих на моделях часових рядів.

У дослідженні було розглянуто шість різних методів пошуку оцінок для заміни відсутніх значень і викидів, і було встановлено, що застосування цих методів не впливає на стаціонарні властивості та поведінкові моделі часових рядів. Було побудовано моделі часових рядів із застосуванням традиційних методів (ARMA), методів глибокого навчання (LSTM) і методу машинного навчання (Facebook Prophet).

Порівняння MAE і RMSE моделей часових рядів показало, що найменші значення можуть бути отримані, коли відсутні значення і викиди замінюються оцінками, оціненими за прогнозними значеннями моделі LSTM для попередніх спостережень.

Наявність пропущених значень і викидів у часових рядах створює труднощі при проведенні статистичного аналізу. Ці перешкоди можуть сильно вплинути на аналіз часових рядів, особливо коли послідовні спостереження в ряді автокорельовані. Відсутні значення і викиди можуть призвести до неправильних висновків про властивості часового ряду, що робить заміну цих

значень на адекватні оцінки важливим завданням при аналізі даних. Загалом, стаття пропонує корисні методи для опрацювання пропущених значень і викидів у часових рядах, які можна застосовувати в різних сферах аналізу даних.

Ця стаття [16] описує дослідження, яке було проведено для перевірки ефективності методу лінійної інтерполяції двох найближчих значень дужок (LIBV) для заповнення відсутніх значень у часових рядах. Для цього було створено імітаційні часові ряди, які були ймовірно еквівалентні типам, що можуть зустрічатися на практиці. Потім було проаналізовано дані про частоту і тривалість епізодів відсутніх значень, щоб визначити ймовірнісні моделі поведінки цих епізодів.

Далі, схему LIBV було використано для заповнення відсутніх значень у змодельованих часових рядах, і було оцінено параметри моделі часового ряду. Отримані результати засвідчили, що схема LIBV може бути ефективною, якщо в даних присутній тільки один домінуючий сезонний компонент. Однак, якщо в даних ймовірно присутні більше одного домінуючого сезонного компонента, то слід використовувати метод з обережністю.

Висновки зі статті вказують на те, що метод лінійної інтерполяції двох найближчих значень дужок (LIBV) може бути ефективним для заповнення відсутніх значень у часових рядах, однак, його використання варто здійснювати з обережністю, враховуючи особливості даних.

Таким чином, стаття підкреслює важливість ретельної оцінки методів заповнення відсутніх значень у часових рядах і необхідність враховувати особливості даних при виборі методу.

Це дослідження [17] було проведене на основі аналізу статистики пожеж у Китаї (з 2003 до 2017 року) з метою визначення регулярності виникнення пожеж та факторів, які на неї впливають. Аналіз даних показав сильні сезонні характеристики місячної частоти пожеж, тому було використано модель SARIMA для прогнозування частоти пожеж. Після оптимізації методом виявлення квантильних викидів і методом інтерполяції аналогічного середнього

значення, оптимізовану модель SARIMA було використано для підгонки прогнозу частоти пожеж.

Лінійна середня інтерполяція (linear mean interpolation) - це метод заповнення пропущених значень у часових рядах, що базується на лінійній апроксимації між двома найближчими ненульовими значеннями ряду. Для кожного пропущеного значення використовуються два найближчих ненульових значення зліва і справа від нього, щоб обчислити середнє значення, яке використовується для заповнення пропуску.

Результати дослідження показали, що оптимізована модель SARIMA є ефективним інструментом для прогнозування, зокрема для частоти пожеж. Значення метрик оцінювання помилок зменшилися, що говорить про те, що оптимізована модель має кращий ефект припасування, ніж модель без обробки викидів. Прогнозовані результати ближчі до фактичних значень, що вказує на кращий ефект прогнозування.

Висновки з цього дослідження вказують на значущість використання оптимізованих моделей SARIMA, зокрема для прогнозування частоти виникнення пожеж у Китаї. Вказується на важливість використання оптимізованих методів прогнозування часових рядів для поліпшення точності та ефективності прогнозування.

## **1.6 Виявлення аномалій у часових рядах**

Дана стаття [18] описує методи ефективного виявлення аномальних споживачів електроенергії та аналізу просторового розподілу і чинників впливу аномального споживання електроенергії в містах за допомогою процедури розкладання часових рядів на основі сезонно-трендового розкладання на основі Loess (STL) і виявлення викидів, а також Geodetector.

У цій статті використовується метод розкладання часових рядів на основі сезонно-трендового розкладання на основі Loess (STL). Цей метод дає змогу розкласти часовий ряд на кілька складових, включно із сезонною, трендовою та

залишковою складовими. Спочатку відбувається розкладання часового ряду на сезонну і трендову складові. Потім із трендової складової віднімається сезонна, щоб отримати залишкову складову, яка містить інформацію про аномальні події в даних.

Після застосування методу STL був проведений аналіз залишкових даних для виявлення викидів, які можуть вказувати на аномальне споживання електроенергії. Для цього було використано метод *box line graph*, за яким і відбувається детекція викидів. Для виявлення аномалій встановлюються ліміти, які визначають, які значення необхідно розглядати як аномальні точки споживання електроенергії клієнтами.

Крім того, у статті було проведено аналіз між просторовою варіацією та міськими функціями, пов'язаними зі споживанням електроенергії в міських районах. Для цього було використано метод *Geodetector*, який дає змогу досліджувати цей взаємозв'язок.

Результати дослідження показали, що ступінь аномального споживання електроенергії в міських районах має просторову неоднорідність і пов'язана з географічним положенням. Аномальні споживачі електроенергії здебільшого розташовані в районах із сильно змішаними житловими, комерційними та розважальними функціями в місті.

Ці результати можуть стати довідковою базою і теоретичною основою для виявлення аномального споживання електроенергії та постановки на охорону пристроїв для розкрадання електроенергії в електромережі. Це може призвести до більш ефективного використання електроенергії, безпечної роботи електромереж і сталого розвитку міст.

Таким чином, у статті було використано методи STL-декомпозиції, метод *box line graph* і *Geodetector* для виявлення аномального споживання електроенергії та аналізу просторової варіації та міських функцій, пов'язаних із ним. Приклад, описаний у статті, показує, що можна використовувати метод розкладання часового ряду на основі STL для виявлення аномалій, зокрема у споживанні електроенергії. Аналіз результатів, отриманих під час дослідження,

показує, що метод STL, метод box line graph, заснований на залишках після STL-декомпозиції, дає змогу виявляти викиди. Таким чином, дослідження показує, що метод STL є ефективним інструментом для виявлення аномалій, зокрема у споживанні електроенергії, що може бути корисним для забезпечення безпечної роботи електромереж, підвищення якості споживання електроенергії споживачами та сталого розвитку міст.

Стаття [19] розповідає про те, що під час аналізу часових рядів необхідно виявляти та видаляти викиди, також відомі як аномалії. Ці спостереження можуть бути помилковими або неінформативними і можуть вплинути на точність аналізу даних. Одним зі способів виявлення аномалій є процедура декомпозиції сезонного тренда на основі LOESS, також відома як STL. Цей метод розділяє часовий ряд на три частини: сезонну, трендову і залишкову, і аналізує відхилення залишків, щоб виявити аномалії.

Перевагою використання методу STL є його простота та надійність, а також можливість опрацювання різних ситуацій. Усі аномалії можуть бути інтуїтивно зрозумілі та інтерпретовані.

Однак, є й недоліки. Головним мінусом методу є його жорсткі можливості налаштування. Крім порога і, можливо, довірчого інтервалу, аналітик не може зробити багато змін. Наприклад, якщо аналізуються дані користувачів на сайті, який був закритий для публіки, а потім раптово був відкритий, необхідно окремо відстежувати аномалії, що виникають до і після періодів запуску.

Висновки зі статті полягають у тому, що аналітики мають бути уважними до наявності викидів у даних часових рядів і використовувати різні методи для їх виявлення та усунення. STL є одним із простих і надійних методів виявлення аномалій за залишками, який може бути використаний для сезонних часових рядів. Однак аналітики також повинні розуміти обмеження цієї техніки. Нарешті, автор статті наголошує на важливості постійного відстеження аномалій і можливості їх відокремлення від періодів запуску, щоб отримати більш точні прогнози. Також не варто забувати, що ще з цими викидами

необхідно проводити подальші маніпуляції, як-от видалення або заміна на відповідні значення, у статті це теж згадується.

### **1.7 Аналіз гібридних моделей**

У статті [20] досліджується проблема прогнозування часових рядів із трендом і сезонними коливаннями, які є нестационарними часовими рядами. Автори розглядають кілька методів прогнозування, включно з моделями Вінтера, декомпозицією, регресією часових рядів, ARIMA та нейронними мережами. У дослідженні використовуються реальні дані про пасажирів авіакомпаній. Автори досліджують ефективність кожного методу прогнозування і хочуть відповісти на питання, чи завжди складніший метод дає кращий прогноз, ніж простіший метод.

Результати дослідження показують, що складніша модель не завжди дає кращий прогноз, ніж простіша модель, особливо на тестових вибірках. Однак, складніша модель FFNN (Feed-Forward Neural Network) завжди дає кращий прогноз на навчальних даних, що може вказувати на проблему перенавчання. Тому простішу модель FFNN слід використовувати під час моделювання та прогнозування трендових і сезонних часових рядів.

Крім того, автори наголошують на можливості подальших досліджень у сфері прогнозування трендів і сезонних часових рядів шляхом комбінування декількох методів прогнозування, включно з методом декомпозиції як попередньої обробки даних у моделі нейронних мереж.

Загалом, дослідження наголошує на необхідності використання найефективніших методів прогнозування для часових рядів із трендами та сезонними закономірностями, а також підкреслює важливість балансу між складністю та ефективністю моделі.

Таким чином, стаття є цікавим дослідженням, яке може бути корисним для економістів і бізнес-аналітиків, які займаються прогнозуванням часових рядів із трендом і сезонними коливаннями. Результати можуть допомогти

вибрати найбільш ефективний метод прогнозування для конкретного набору даних.

Стаття [21] описує розробку швидких одновимірних моделей для прогнозування епідемій туберкульозу в Тайвані. Для моделювання використовувалися дані про щомісячні показники захворюваності на туберкульоз з 2005 року по 2020 рік. Порівнювали різні підходи до моделювання, включно з SARIMA, ETS і гібридні алгоритми SARIMA-ETS. Після оцінювання метрик точності, оптимізовані моделі SARIMA ETS і гібридна модель SARIMA-ETS були відповідно обрані як моделі-кандидати. Дослідження показало, що гібридна модель SARIMA-ETS перевершила ARIMA і ETS у короткостроковому прогнозуванні, а модель ETS показала найкращі результати в довгостроковому прогнозуванні. Варто згадати, що в короткостроковому або середньостроковому прогнозуванні всі моделі демонстрували вищі результати, ніж у довгостроковому. Загалом, стаття показує, що моделювання часових рядів може бути корисним інструментом для прогнозування захворюваності на туберкульоз і планування розподілу ресурсів для боротьби з цією хворобою.

Висновки зі статті такі: гібридна модель SARIMA-ETS і модель ETS показали найкращі результати в прогнозуванні захворюваності на туберкульоз у Тайвані, причому модель ETS була ефективнішою в довгостроковому прогнозуванні.

Стаття [22] розглядає ефективність застосування різних моделей для прогнозування часових рядів, пов'язаних із пандемією COVID-19. Автори вивчають три моделі: авторегресійне інтегроване ковзне середнє (ARIMA), моделі штучних нейронних мереж (ANN) і моделі помилок, тренду і сезонності (ETS). Автори порівнюють ефективність цих моделей, використовуючи реальні дані за період із січня 2020 року по січень 2021 року, що відображають дві хвилі пандемії.

Автори доходять висновку, що моделі ETS і ARIMA виявляються більш ефективними для прогнозування часових рядів COVID-19, ніж ANN і гібридні

моделі. Також автори дослідження показують, що ідеальним рішенням було б використання тільки однієї моделі ETS або ARIMA для прогнозування часового ряду COVID-19, а не складної гібридної моделі, що об'єднує кілька моделей. Це може скоротити час і ресурси, що витрачаються на аналіз даних і прогнозування тенденцій пандемії.

Загалом, стаття демонструє, що моделі ETS і ARIMA можуть бути ефективними для прогнозування часових рядів COVID-19, і що використання однієї з цих моделей може бути більш ефективним, ніж складні гібридні моделі. Цей підхід може допомогти прискорити аналіз даних і прогнозування тенденцій пандемії, що має велике значення для ухвалення рішень у медичному та економічному секторах. Однак, слід зазначити, що результати дослідження можуть бути унікальними для часового ряду COVID-19 і можуть не переноситися на інші типи часових рядів.

### **1.8 Автоматизація часових рядів**

Дана стаття [23] розглядає проблему прогнозування часових рядів з використанням машинного навчання, глибокого навчання і AutoML. В експериментах використовувалися дані про реальні ціни на криптовалюти. У результаті експериментів було встановлено, що AutoML для часових рядів перебуває на стадії розроблення і потребує додаткового вивчення, щоб стати життєздатним рішенням. Однак наведені методи можуть слугувати хорошим початковим кроком для досягнення задовільної точності в прогнозуванні часових рядів. Водночас, моделі машинного навчання та глибокого навчання, розроблені вручну, продемонстрували кращі результати.

З цієї статті можна зробити висновок, що в галузі прогнозування часових рядів важливо використовувати різні методи машинного навчання та глибокого навчання для досягнення найкращих результатів. Незважаючи на те, що AutoML може надати автоматизований підхід до створення моделей, він все ще потребує додаткового вивчення та розробки, щоб досягти

конкурентоспроможності з ручним розробленням моделей. Крім того, дані про ціни на криптовалюти можуть бути використані як набір даних для прогнозування часових рядів у цій галузі.

У даній статті [24] розглядається модель ARIMA та метод Auto ARIMA, який автоматизує процес вибору оптимальної моделі ARIMA.

ARIMA - це статистична модель, яка використовує дані часових рядів для прогнозування майбутніх тенденцій. Вона передбачає майбутні значення на основі минулих значень, наприклад, може використовуватися для прогнозування цін на акції або прибутку компанії.

Auto ARIMA - це метод, який автоматизує процес вибору оптимальної моделі ARIMA. Він зіставляє безліч моделей в іншому порядку, щоб вибрати ту, що дає найвищу логарифмічну правдоподібність і потребує найменшого AIC. Цей метод економить час і знижує ризик людської помилки, не вимагає від користувача глибокого розуміння статистики та теорії.

Однак, метод Auto ARIMA також має свої недоліки. Використовуючи лише один критерій (AIC), сліпо покладаємося на обрану модель, не знаючи, наскільки добре працюють інші моделі. Також, час обробки значно збільшується, якщо намагатися підігнати складні моделі.

Загалом, метод Auto ARIMA може бути корисним для тих, хто хоче автоматизувати процес вибору моделі ARIMA і не має достатнього досвіду в статистиці. Однак, він не є єдиним методом, інші методи також можуть дати хороші результати. Важливо оцінювати не тільки найвищу логарифмічну правдоподібність і AIC, а й інші критерії, такі як точність прогнозування на тестовому наборі даних.

Таким чином, Auto ARIMA - це корисний інструмент для швидкого вибору моделі ARIMA, що дає змогу автоматизувати процес, але необхідно враховувати його обмеження і не покладатися лише на один метод під час прогнозування часових рядів.

## 1.9 Система підтримки ухвалення рішень із прогнозуванням

Система підтримки ухвалення рішень із прогнозуванням (Forecasting Decision Support System) являє собою комп'ютерну систему, яка допомагає організаціям і окремим особам ухвалювати рішення, ґрунтуючись на прогнозах і передбаченнях майбутніх подій і тенденцій. Вона використовує статистичні методи, аналітичні моделі та алгоритми машинного навчання для аналізу та інтерпретації історичних даних і визначення майбутніх тенденцій.

Forecasting Decision Support System допомагає прогнозувати різні параметри, як-от продажі, попит на товари та послуги, фінансові показники, виробничі обсяги та інші фактори, що можуть впливати на ухвалення рішень в організації. Вона дає змогу аналізувати минулі дані, ідентифікувати патерни та тренди, а потім використовувати цю інформацію для створення прогнозів і моделей, які допомагають організаціям ухвалювати більш усвідомлені рішення.

Переваги використання системи підтримки ухвалення рішень з прогнозуванням включають підвищення точності прогнозів, поліпшення планування ресурсів і операцій, оптимізацію процесів ухвалення рішень, поліпшення управління ризиками та забезпечення конкурентної переваги на ринку. Ця система може бути використана в різних галузях і функціональних сферах, включаючи фінанси, логістику, маркетинг, виробництво та інші.

Загалом, Forecasting Decision Support System забезпечує організаціям та особам інструменти й аналітичні можливості для ухвалення більш обґрунтованих та поінформованих рішень, що ґрунтуються на прогнозах майбутніх подій і тенденцій.

У даній дослідницькій роботі [25] представлено систему підтримки ухвалення рішень, що використовує методи Вінтерса, простого експоненціального згладжування та регресійного статистичного аналізу з використанням запропонованого генетичного алгоритму для генерації операційного прогнозу.

Прогнозування з використанням одного статистичного методу показало обмеження в наданні точного прогнозу. Використання системи підтримки

ухвалення рішень необхідне для забезпечення точного прогнозування, особливо для складних середовищ, які мають різні чинники прогнозування. Як описано в даній статті, пропонується використовувати модифікований генетичний алгоритм для комбінування статистичних методів і об'єднання дій з базовим прогнозом, що дає змогу отримати більш точні прогнози. Зокрема генетичний алгоритм шукає найкращу комбінацію ваг між методами, яка мінімізує середньоквадратичну помилку. Обраний приклад базується на товарах народного споживання, які мають різні типи часових рядів. Порівняння отриманих результатів показує вищу точність, ніж при використанні традиційних методів. У пропонуваному рішенні можна включити інші комбінації прогностичних методів для підвищення точності прогнозування.

### **1.10 Висновок по першому розділу**

Аналіз літератури показав, що прогнозування часових рядів є важливою галуззю досліджень, яка знаходить застосування в різних сферах, від економіки до охорони здоров'я. Під час аналізу було розглянуто різні методи, які використовуються в практичній частині роботи, такі як методи ARIMA, експоненціальне згладжування та гібридні моделі.

Один із найбільш часто використовуваних методів - ARIMA - показав адекватність у прогнозуванні часових рядів. Він є універсальним і може бути використаний у різних сферах життя, де необхідно прогнозувати зміни в часі. Теж саме стосується й експоненціального згладжування, яке також є ефективним методом прогнозування часових рядів. Кожна з цих моделей має свої плюси і мінуси, і що вони можуть по-різному змодельовати потрібний прогнозний результат. Вони не є взаємовиключним випадком. Тож важливо враховувати контекст даних, що моделі можуть по-різному побудувати прогнози на них.

Однак для досягнення більш високої точності моделі необхідно проводити попередню обробку даних. Це дає змогу поліпшити точність моделі,

зокрема, за рахунок виявлення аномалій і пропущених значень. Застосування методу STL для виявлення аномалій і різні методи заповнення пропущених значень може значно поліпшити ту саму точність прогнозування.

Крім того, оцінювання якості прогнозування є важливим етапом дослідження. Різні методи оцінки, такі як MAPE, можуть допомогти в оцінці якості моделі. Важливо зазначити, що іноді краще використовувати простішу модель, ніж складнішу, яка може не завжди давати кращий результат.

Гібридні моделі - це ще одна область досліджень у прогнозуванні часових рядів. Вони поєднують у собі переваги різних моделей і дають змогу досягати вищої точності прогнозування.

Автоматизація процесу прогнозування часових рядів є одним з основних моментів дослідження. Вона дає змогу проводити автоматизований аналіз, економить час, і знижує ризик через суб'єктивний підхід аналітика. Однак ручний аналіз може показувати все ще більш конкурентоспроможний результат.

Було розглянуто систему підтримки ухвалення рішень із прогнозуванням, що надає організаціям та окремим особам інструменти й аналітичні можливості для ухвалення більш ґрунтовних та поінформованих рішень, що ґрунтуються на прогнозах і передбаченнях майбутніх подій і тенденцій.

Загалом, прогнозування часових рядів - це важлива галузь досліджень, яка має практичне застосування в різних галузях, і її розвиток та дослідження є актуальним та перспективним напрямом. Зрештою, наш аналіз літератури допоміг нам зрозуміти, які методи та підходи можуть бути найефективнішими у прогнозуванні часових рядів у різних сферах застосування, зокрема в нашій роботі. Правильний вибір методу прогнозування та його точність може мати значний вплив на ухвалення рішень у різних сферах життя, тому подальші дослідження в цій галузі є важливими.

## **РОЗДІЛ 2. ФОРМАЛІЗАЦІЯ МОДЕЛЕЙ, МЕТОДІВ ТА ТЕХНОЛОГІЙ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ДЛЯ ПРИЙНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ**

### **2.1 Особливості моделей часових рядів при прогнозуванні**

Часовий ряд - це послідовність даних, виміряних у різні моменти часу, які можна використовувати для аналізу та прогнозування. Часові ряди є одним з основних типів даних в економетриці, статистиці, фінансах, інженерії, науці про клімат та інших галузях. Часовий ряд складається з двох змінних: часової змінної та змінної, яка характеризує величину, що нас цікавить (наприклад, ціну товару, обсяг продажів, температуру тощо). Часова змінна являє собою послідовність моментів часу, в які вимірювалися значення змінної [26]. Важливо зазначити, що часовий ряд має бути впорядкований за часом, тобто спостереження мають бути впорядковані за зростанням часової змінної.

Приклади часових рядів включають в себе щоденні дані про ціни на нафту, щомісячні дані про кількість пасажирів, що перевозяться авіакомпаніями, щорічні дані про випуск автомобілів і багато іншого. Однією з ключових особливостей часових рядів є наявність залежності між вимірами. Це означає, що значення в один момент часу можуть впливати на значення в майбутньому або минулому. Крім того, часові ряди можуть містити шум, що робить їх складнішими для аналізу та прогнозування.

Тренд часового ряду - це загальний напрям зміни значень спостережуваної змінної в часі. Він являє собою довгострокову тенденцію в даних, яка може бути зумовлена різними факторами, включно зі змінами в економічних умовах, технологічними проривами, демографічними змінами та іншими соціально-економічними чинниками.

Тренд може бути лінійним або нелінійним. Лінійний тренд характеризується рівномірною зміною значень змінної з часом, у той час як нелінійний тренд може бути криволінійним і містити зміни швидкості або напрямку зміни. Тренд може бути позитивним або негативним, залежно від

того, збільшуються чи зменшуються значення змінної з часом. Також тренд може бути стаціонарним, коли значення змінної коливаються навколо постійного рівня без явної тенденції. Про це буде сказано трохи пізніше.

Крім того, тренд може бути локальним або глобальним [27]. Локальний тренд у часовому ряді - це зміна значень змінної в певному періоді часу, що відбувається у висхідному або низхідному напрямі всередині більш загального глобального тренда. Своєю чергою, глобальний тренд являє собою загальну тенденцію зміни значень змінної протягом усього періоду спостереження, незалежно від локальних коливань. Водночас всередині глобального тренду може бути кілька локальних трендів. Наприклад, протягом року продажі можуть спочатку зростати, потім знижуватися, а потім знову починати зростати. У цьому випадку локальний тренд буде являти собою зміну продажів протягом окремих місяців або кварталів.

Сезонність часового ряду - це періодичні коливання в часовому ряді, які відбуваються в певну пору року або в певних інтервалах часу. Часовий ряд із сезонністю може бути представлений у вигляді повторюваних циклів, які повторюються в певний час або в певних інтервалах часу.

Сезонність може проявлятися в різних галузях, наприклад, у фінансах, маркетингу, економіці, кліматології та інших. Наприклад, продажі літніх товарів можуть бути вищими в літні місяці, а продажі зимових товарів - у зимові. Частота сезонності може бути річною, квартальною, місячною, тижневою або навіть денною, залежно від того, які періоди часу аналізуються. Визначення сезонності в часовому ряді є важливим кроком під час аналізу часових рядів, тому що це може допомогти в розумінні структури ряду.

Цикл часового ряду - це періодичні коливання в часовому ряді, що відбуваються в певних інтервалах часу, але не обов'язково з постійною частотою, як це буває за сезонності. Циклічні коливання можуть бути тривалішими, ніж сезонні коливання, і мати ширший діапазон змін. У деяких випадках циклічні коливання можуть спостерігатися протягом кількох років або навіть десятиліть.

Випадковий шум (або випадкова помилка) у часовому ряді - це непередбачувані та випадкові коливання в даних, які не можуть бути пояснені сезонністю, трендом або циклічними факторами. Ці коливання можуть статися через випадкові фактори, як-от помилки вимірювання, випадкові події або незначні зміни зовнішніх умов.

Випадковий шум може бути видалений з даних шляхом застосування різних методів, таких як згладжування або фільтрація даних. Однак не завжди доцільно видаляти випадковий шум, оскільки він може містити важливу інформацію про непередбачувані зміни, які можуть вплинути на результати аналізу. Зокрема через його виділення з ряду, можна за допомогою певних маніпуляцій провести виявлення аномалій.

Візуалізація часового ряду - це один із перших і важливих кроків в аналізі часових рядів. Вона дає змогу наочно уявити зміни значення ряду в часі та виявити особливості його поведінки. Для візуалізації часового ряду використовують графіки. Вісь X на графіку зазвичай відображає часовий період, а вісь Y - значення ряду. Одна з найпоширеніших форм візуалізації часового ряду - це лінійний графік, на якому точки з'єднані прямими лініями. Також можуть використовуватися інші види графіків, наприклад, стовпчасті, точкові, сплайнові та інші.

Візуалізація часового ряду дає змогу виявити різні особливості поведінки ряду, такі як тренди, сезонність, цикли, викиди та пропуски значень. Це допомагає визначити, які методи аналізу та прогнозування часового ряду слід застосовувати. Загалом, візуалізація часового ряду є потужним інструментом для первинного аналізу часових рядів і допомагає дослідникам отримати глибше розуміння їхніх властивостей і поведінки.

Аналіз і прогнозування часових рядів - це важні методи статистичного аналізу, що дозволяють вивчати зміни явищ та процесів у часі. Часові ряди - послідовність значень змінної, спостережених в різні моменти часу, які можуть бути використані для аналізу та прогнозування майбутніх тенденцій. Методи аналізу та прогнозування часових рядів широко застосовуються у різних

галузях, таких як економіка, фінанси, маркетинг, виробництво і охорона здоров'я, для вивчення та передбачення різних параметрів, таких як обсяги продажів, ціни на товари, фінансові показники, виробничі показники та показники здоров'я пацієнтів.

Автокореляція - це статистичний показник, який описує ступінь залежності між значеннями в одному й тому самому часовому ряді. Вона показує, наскільки кожне значення в ряду пов'язане з попередніми значеннями [28]. Формально, автокореляція визначається як кореляція між одним і тим самим часовим рядом, але з різним лагом. Іншими словами, це показник кореляції між значенням часового ряду в момент часу  $t$  і значеннями цього ж ряду в минулому або майбутньому часі  $t + k$  (де  $k$  - лаг). Якщо автокореляція висока, то це означає, що значення часового ряду в минулому сильно впливають на значення в сьогоденні та майбутньому. Автокореляція може допомогти у визначенні структури та залежностей у часовому ряді, наприклад, для визначення сезонності або циклічності в даних, а також у визначенні оптимальних параметрів для моделей часових рядів, таких як модель ARIMA.

Однією з важливих властивостей часових рядів є стаціонарність. Стаціонарний часовий ряд має постійне середнє значення, постійну дисперсію та автокореляційну функцію, яка не залежить від часу [29]. Для перетворення ряду в стаціонарний вигляд застосовуються спеціальні методи, наприклад, диференціювання. Стаціонарність часового ряду можна перевірити, використовуючи різні статистичні тести, такі як тест Дікі-Фуллера або тест KPSS.

Стаціонарність часового ряду важлива для прогнозування, оскільки моделі, засновані на стаціонарних рядах, відносно легше передбачити. Важливо зазначити, що навіть якщо часовий ряд стаціонарний, це не гарантує його точність в аналізі та прогнозуванні. Тому перед застосуванням будь-яких методів необхідно переконатися в їхній застосовності та точності на конкретному ряді даних.

## 2.2 Моделювання часових рядів

Моделювання часових рядів є важливим етапом в аналізі часових рядів, який дає змогу створити математичну модель, що описує закономірності у зміні значень ряду в часі. Ця модель може бути використана для прогнозування майбутніх значень ряду та розуміння факторів, що впливають на зміну значень.

Існує безліч методів [30] прогнозування часових рядів, кожен з яких має свої переваги та обмеження. Наведемо деякі методи прогнозування часових рядів, зокрема також ті, які будуть використовуватися в практичній частині дослідження.

### 2.2.1 Просте експоненціальне згладжування

Модель простого експоненціального згладжування (Simple Exponential Smoothing) є одним із варіантів експоненціального згладжування, що використовуються для прогнозування часових рядів. Вона підходить для рядів без тренду та сезонності.

Просте експоненціальне згладжування базується на припущенні, що поточне значення часового ряду залежить від попередніх значень, причому ваги спостережень зменшуються експоненціально в міру віддалення від поточного моменту часу. Це означає, що новіші значення мають більшу вагу, ніж старіші значення.

Основна ідея моделі простого експоненціального згладжування полягає в тому, що прогноз майбутнього значення ряду розраховується шляхом зваженої суми останнього спостереження і попереднього прогнозу. Ваги спостережень визначаються коефіцієнтом згладжування (smoothing factor), який позначається як  $\alpha$  (alpha).

Коефіцієнт згладжування  $\alpha$  набуває значень від 0 до 1 і визначає, яку частку ваги віддають останньому спостереженню і попередньому прогнозу. Якщо  $\alpha$  близьке до 1, то останнє спостереження матиме більшу вагу і прогноз буде більш чутливий до останніх змін у ряді. Якщо  $\alpha$  близьке до 0, то прогноз

буде менш чутливий до останніх спостережень і більш стійкий до шумів і випадкових флуктуацій.

Формула (1) для розрахунку прогнозу на наступний період (T+1) у моделі простого експоненціального згладжування [31]:

$$F(T+1) = \alpha * Y(T) + (1-\alpha) * F(T), \quad (1)$$

де  $F(T)$  - прогноз на поточний період,

$Y(T)$  - значення ряду на поточний період.

Спочатку, для розрахунку першого прогнозу  $F(1)$  потрібне початкове значення ( $Y(0)$ ) або початкове середнє значення ряду.

Однією з переваг моделі простого експоненціального згладжування є її простота у використанні та розумінні. Також вона дає змогу швидко розраховувати прогнози, що особливо корисно для короткострокових прогнозів без складних трендів і сезонності.

Однак у моделі простого експоненціального згладжування є й недоліки. Вона не враховує зміни тренду або сезонності в даних, тому не підходить для рядів з яскраво вираженими трендами і сезонними коливаннями. Також вона може бути менш точною в прогнозуванні, як порівняти зі складнішими моделями, які враховують більше чинників та особливостей часового ряду.

### 2.2.2 Подвійне експоненціальне згладжування

Подвійне експоненціальне згладжування (Double Exponential Smoothing) є методом прогнозування часових рядів, який враховує як рівень (Level), так і тренд (Trend) у даних. Він є розширенням простого експоненціального згладжування і призначений для аналізу часових рядів зі зростаючим або спадним трендом.

Основна ідея подвійного експоненціального згладжування полягає у використанні двох компонент для прогнозування:

Рівень: являє собою згладжене значення ряду на поточний період часу.

Тренд: являє собою зміну ряду на кожному періоді часу.

Подвійне експоненціальне згладжування складається з двох (2)(3) основних рівнянь [32]:

$$S(t) = \alpha * Y(t-1) + (1 - \alpha) * (S(t-1) + b(t-1)), \quad (2)$$

$$b(t) = \beta * (S(t) - S(t-1)) + (1 - \beta) * b(t-1), \quad (3)$$

де  $S$  - позначає згладжене значення

$Y$  - позначає часовий ряд

$t$  - позначає період часу часового ряду у і набуває значень від 1 до  $n$

$\alpha$  - коефіцієнт згладжування для згладженого значення ( $0 < \alpha \leq 1$ ).

$\beta$  - коефіцієнт згладжування для тренда ( $0 < \beta \leq 1$ ).

Прогноз на наступний період (4) розраховується з використанням рівня і тренда:

$$F(t+m) = S(t) + m * b(t), \quad (4)$$

де  $F(t+m)$  - прогноз на період  $t+m$ ,  $m$  - кількість періодів уперед ( $1 \leq m \leq N$ , де  $N$  - позначає горизонт прогнозування)

Подвійне експоненціальне згладжування дає змогу врахувати зростаючий або спадний тренд у даних, що робить його корисним для прогнозування часових рядів із лінійним трендом. Однак, метод не враховує сезонність або інші складні структури в даних, тому він може бути менш точним для рядів зі складнішими патернами або сезонними коливаннями. Для врахування сезонності в даних рекомендується застосовувати методи, такі як потрійне експоненціальне згладжування (Хольта-Вінтерса).

### 2.2.3 Модель Хольта-Вінтерса

Метод Хольта-Вінтерса (Holt-Winters method) - це модель експоненціального згладжування, що розширює метод Хольта, враховуючи не тільки тренд, а й сезонність у даних часових рядів. Він є одним із найпопулярніших методів прогнозування сезонних часових рядів і дає змогу враховувати як тренд, так і сезонні коливання в даних.

Метод Хольта-Вінтерса використовує три компоненти для моделювання ряду:

Рівень: це згладжене значення ряду на поточний період часу, що позначається  $S(t)$ .

Тренд: це швидкість зміни рівня ряду, що позначається  $b(t)$ .

Сезонність: це повторювані коливання ряду всередині кожного сезону, що позначаються  $I(t)$ .

Основні рівняння методу Хольта-Вінтерса [32]:

Рівняння (5) для рівня ( $S$ ):

$$S(t) = \alpha * (Y(t) - I(t-L)) + (1 - \alpha) * (S(t-1) + b(t-1)), \quad (5)$$

де  $Y(t)$  - значення ряду на поточний період часу,

$\alpha$  - коефіцієнт згладжування для згладженого значення ( $0 < \alpha \leq 1$ ),

$L$  - позначає період

$I$  - позначає оцінку сезонного компонента

Рівняння (6) для тренда ( $b$ ):

$$b(t) = \beta * (S(t) - S(t-1)) + (1 - \beta) * b(t-1), \quad (6)$$

де  $\beta$  - коефіцієнт згладжування для тренда ( $0 < \beta \leq 1$ ).

Рівняння (7) для сезонності ( $I$ ):

$$I(t) = \gamma * (Y(t) - S(t)) + (1 - \gamma) * I(t-L), \quad (7)$$

де  $\gamma$  - коефіцієнт згладжування для сезонності ( $0 < \gamma \leq 1$ ).

Коефіцієнти згладжування  $\alpha$ ,  $\beta$  і  $\gamma$  дають змогу налаштовувати модель Хольта-Вінтерса залежно від особливостей ряду. Більші значення  $\alpha$ ,  $\beta$  і  $\gamma$  роблять модель більш чутливою до останніх значень ряду, тренду і сезонних коливань відповідно.

Однією з головних переваг методу Хольта-Вінтерса є його здатність враховувати і прогнозувати сезонність, що робить його особливо корисним для рядів з явно вираженими сезонними патернами. Він також може моделювати тренди в даних, що змінюються. Однак у методу Хольта-Вінтерса є і деякі обмеження. Наприклад, він вимагає досить довгого і стабільного історичного ряду для точного моделювання сезонності.

Загалом, метод Хольта-Вінтерса є потужним інструментом для прогнозування сезонних часових рядів з урахуванням тренду і сезонності. Він може бути застосований у різних галузях, включно з економікою, фінансами, продажами, виробництвом та іншими, де сезонні коливання відіграють важливу роль у передбаченні майбутніх значень.

#### 2.2.4 AR модель

AR (авторегресійна) модель - це одна з основних моделей, що використовуються для аналізу та прогнозування часових рядів. Вона передбачає, що значення часового ряду в певний момент часу залежить від його попередніх значень і випадкової помилки. В AR-моделі використовуються тільки минулі значення ряду для прогнозування його майбутніх значень, тому вона називається "авторегресійною".

AR-модель може бути представлена як рівняння (8), де поточне значення часового ряду є функцією його попередніх значень і випадкової помилки. Це рівняння має такий вигляд [33]:

$$Y(t) = c + \phi_1 Y(t-1) + \phi_2 Y(t-2) + \dots + \phi_p Y(t-p) + e(t), \quad (8)$$

де  $Y(t)$  - поточне значення часового ряду;

$c$  - константа;

$\phi_1, \phi_2, \dots, \phi_p$  - коефіцієнти авторегресії, які відображають вплив попередніх значень на поточне значення;

$Y(t-1), Y(t-2), \dots, Y(t-p)$  - попередні значення часового ряду;

$e(t)$  - випадкова помилка або шум.

Коефіцієнти  $\phi_1, \phi_2, \dots, \phi_p$  визначають за допомогою методу найменших квадратів, який дає змогу оцінити оптимальні значення коефіцієнтів, що мінімізують суму квадратів помилок моделі. Що вище значення  $p$ , то більше минулих значень часового ряду використовується в моделі, що може поліпшити точність прогнозування, але також може збільшити складність моделі.

AR-модель дає змогу оцінити як короткострокові, так і довгострокові тенденції часового ряду. Крім того, її можна розширити до ARMA (авторегресійно-ковзного середнього) моделі, яка враховує не тільки минулі значення ряду, а й помилки, що відбуваються в процесі моделювання.

Важливим аспектом при використанні AR-моделі є перевірка її адекватності та відповідності статистичним критеріям. Для цього використовуються різні тести, такі як тест Дарбіна-Вотсона та тест Льюнга-Бокса, які дають змогу оцінити, наскільки добре модель описує дані та не має систематичних помилок.

Однак, при використанні AR-моделі необхідно враховувати кілька обмежень. По-перше, вона припускає, що минулі значення часового ряду можуть повністю описати його майбутню поведінку, що не завжди є вірним у реальному житті. По-друге, вона може виявитися неадекватною, якщо в даних присутня сильна сезонність або нелінійні зміни.

Таким чином, AR-модель є важливим інструментом для аналізу та прогнозування часових рядів, що може бути корисним у різних галузях, таких як економіка, фінанси, кліматологія та інші. Однак, перед використанням цієї моделі необхідно ретельно оцінити її адекватність і враховувати можливі обмеження.

### 2.2.5 MA модель

MA (ковзного середнього) модель - це ще одна модель, що використовується для аналізу та прогнозування часових рядів. Вона передбачає, що значення часового ряду в певний момент часу залежить від випадкових помилок, які відбуваються в процесі моделювання і залежать від попередніх значень цих помилок і випадкового шуму.

MA-модель може бути представлена як рівняння (9), де поточне значення часового ряду є функцією попередніх випадкових помилок і випадкового шуму. Це рівняння має такий вигляд [34]:

$$Y(t) = c + e(t) + \theta_1 e(t-1) + \theta_2 e(t-2) + \dots + \theta_q e(t-q), \quad (9)$$

де  $Y(t)$  - поточне значення часового ряду;

$c$  - середнє значення часового ряду;

$e(t), e(t-1), e(t-2), \dots, e(t-q)$  - випадкові помилки;

$\theta_1, \theta_2, \dots, \theta_q$  - коефіцієнти ковзної середньої, які відображають вплив попередніх випадкових помилок на поточне значення.

Аналогічно до моделі AR, коефіцієнти  $\theta_1, \theta_2, \dots, \theta_q$  визначають за допомогою методу найменших квадратів, який дає змогу оцінити оптимальні значення коефіцієнтів, що мінімізують суму квадратів помилок моделі.

МА-модель також може бути розширена до моделі ARMA, яка поєднує в собі авторегресійну та ковзну середню моделі.

У моделі МА кожне значення ряду  $Y(t)$  може бути розглянуте як зважене ковзне середнє останніх помилок прогнозу. Водночас зверніть увагу, що моделі ковзного середнього не слід плутати зі згладжуванням ковзного середнього. На відміну від МА-моделі, згладжування ковзного середнього використовується для оцінювання циклу тренду минулих значень ряду. Згладжування ковзного середнього дає змогу видалити шум та інші випадкові коливання в даних, щоб виділити основну тенденцію ряду.

Важливим аспектом при використанні МА-моделі є перевірка її адекватності та відповідності статистичним критеріям, таким як тест Дарбіна-Вотсона та тест Льюнга-Бокса.

Одним із недоліків моделі МА є те, що вона не може враховувати довгострокові тенденції часового ряду. Крім того, якщо модель містить багато коефіцієнтів ковзної середньої, це може призвести до перенавчання і зниження точності прогнозування.

Незважаючи на деякі обмеження, модель МА широко використовується в різних галузях, таких як економіка, фінанси, метеорологія та інші, і може бути корисною для аналізу та прогнозування часових рядів.

### 2.2.6 ARMA модель

ARMA (авторегресійно-ковзаюче середнє) модель - це комбінація моделей AR (авторегресія) і MA (ковзаюче середнє), що дає змогу врахувати як автокореляцію, так і кореляцію помилок моделі. Вона використовується для аналізу та прогнозування часових рядів, які можуть містити як автокореляцію, так і кореляцію помилок.

ARMA-модель може бути представлена таким рівнянням (10) [35]:

$$Y(t) = \theta_1 Y(t-1) + \theta_2 Y(t-2) + \dots + \theta_p Y(t-p) + \alpha_1 e(t-1) + \alpha_2 e(t-2) + \dots + \alpha_q e(t-q) + e(t), \quad (10)$$

де  $Y(t)$  - поточне значення часового ряду;

$\theta_1, \theta_2, \dots, \theta_p$  - коефіцієнти авторегресії, які відображають вплив попередніх значень на поточне значення;

$\alpha_1, \alpha_2, \dots, \alpha_q$  - коефіцієнти ковзної середньої, які відображають вплив попередніх помилок на поточне значення;

$Y(t-1), Y(t-2), \dots, Y(t-p)$  - попередні значення часового ряду;

$e(t-1), e(t-2), \dots, e(t-q)$  - попередні значення помилок;

$e(t)$  - випадкова помилка або шум.

Коефіцієнти моделі ARMA також можуть бути оцінені за допомогою методу найменших квадратів. Однак, вибір оптимальних значень  $p$  і  $q$ , тобто кількість лагів, що використовуються для опису часового ряду, може бути складним завданням, що вимагає ретельного аналізу даних.

ARMA-модель дає змогу оцінити як короткострокові, так і довгострокові тенденції часового ряду. На відміну від AR-моделі, яка враховує тільки автокореляцію, ARMA-модель дає змогу враховувати кореляцію помилок, що може поліпшити точність прогнозування.

Важливим аспектом під час використання ARMA-моделі є перевірка її адекватності та відповідності статистичним критеріям. Тут можна згадати ті самі тести, які було описано раніше, як-от тест Дарбіна-Вотсона і тест Льюнга-Бокса.

ARMA-модель широко використовується в економіці, фінансах, погоді, маркетингу та інших галузях, де необхідно аналізувати і прогнозувати часові ряди. Її можна використовувати для прогнозування майбутніх значень часового ряду, а також для оцінювання ефектів маркетингових кампаній, економічних подій, зміни клімату та інших чинників, які можуть вплинути на часовий ряд.

### 2.2.7 ARIMA модель

ARIMA модель - це розширення моделі ARMA, що враховує як автокореляцію, так і нестационарність часових рядів, тобто тенденцію і сезонність [36].

Однак, ARIMA модель додає ще один параметр  $d$  - порядок інтеграції, який визначає кількість разів, яку потрібно зробити різниці, щоб зробити ряд стаціонарним. Різниця - це різниця між поточним значенням ряду та попереднім значенням ряду.

Таким чином, ARIMA модель містить у собі три параметри:  $p$  - порядок авторегресії,  $d$  - порядок інтеграції та  $q$  - порядок ковзного середнього.

ARIMA модель може бути використана для аналізу та прогнозування часових рядів, які можуть містити як автокореляцію, так і нестационарність. Нестационарність часового ряду може бути спричинена наявністю тренда, циклу або сезонності в даних. ARIMA модель дає змогу оцінити як короткострокові, так і довгострокові тенденції часового ряду та враховує зміни, що можуть відбутися в часі.

ARIMA модель може бути застосована до широкого спектра даних, таких як фінансові ринки, погодні дані, соціальні та економічні показники тощо. Однак, вибір оптимальних значень  $p$ ,  $d$  і  $q$  може бути складним завданням, що вимагає досвіду та експертизи. Зазвичай для підбору оптимальних значень параметрів використовують методи автоматичного вибору моделі, як-от інформаційний критерій Акаїке (AIC) і критерій Шварца (BIC).

AIC і BIC - це методи, які дають змогу оцінити якість моделі та порівняти її з іншими моделями. Вони враховують не тільки точність моделі, а й її складність. Що менше значення AIC або BIC, то краща модель. Однак, вибір оптимальної моделі має ґрунтуватися не тільки на значенні AIC або BIC, а й на експертній думці та розумінні специфіки даних.

ARIMA модель може бути також розширена до SARIMA моделі, яка враховує сезонність у даних. SARIMA модель містить у собі додаткові параметри, які визначають сезонність у даних і дають змогу моделювати короткострокові та довгострокові тенденції.

Загалом, ARIMA модель є потужним інструментом для аналізу та прогнозування часових рядів, які можуть містити як автокореляцію, так і нестационарність. Однак, для досягнення точних прогнозів і вибору оптимальних значень параметрів, потрібен досвід і експертиза в галузі аналізу часових рядів.

### 2.2.8 SARIMA модель

Модель SARIMA (Seasonal Autoregressive Integrated Moving Average) - це розширення моделі ARIMA, яке враховує сезонність у даних.

Для створення сезонної моделі ARIMA необхідно додати в модель ARIMA додаткові умови, що враховують сезонність даних. Таке уточнення може бути записано таким чином [37]: ARIMA (p,d,q) (P,D,Q)S

SARIMA модель містить три параметри: p - порядок авторегресії, d - порядок інтеграції, q - порядок ковзного середнього. Крім того, модель SARIMA також включає параметри P, D, Q і S, які відображають порядок сезонної авторегресії, порядок інтеграції, порядок ковзного середнього та довжину сезонного періоду для сезонних різниць ряду.

SARIMA модель дає змогу враховувати як довгострокові, так і короткострокові тенденції в даних, а також сезонність. Ця модель може бути

використана для аналізу та прогнозування часових рядів, які містять як автокореляцію, так і сезонність.

Вибір оптимальних значень  $p$ ,  $d$ ,  $q$ ,  $P$ ,  $D$  і  $Q$  для SARIMA моделі може бути складним завданням і вимагає ретельного аналізу та підбору параметрів. Для визначення оптимальних значень параметрів SARIMA моделі може бути використано метод Grid Search, що полягає в переборі всіх можливих комбінацій значень параметрів і виборі тієї, яка дає найкращі результати.

Крім того, SARIMA модель може бути поліпшена за допомогою включення екзогенних змінних. Екзогенні змінні - це змінні, які можуть впливати на значення цільового часового ряду, але не залежать від нього. Включення екзогенних змінних у SARIMA модель може збільшити її точність і передбачувальну силу.

SARIMA модель може бути застосована в різних галузях, як-от економіка, фінанси, метеорологія, маркетинг, виробнича і технічна сфери, де необхідно аналізувати та прогнозувати часові ряди із сезонністю та автокореляцією.

### **2.3 Параметри моделі ARIMA**

Для застосування методу ARIMA необхідно налаштувати його параметри. Основними параметрами ARIMA є:

- Параметр авторегресії ( $p$ ): визначає кількість попередніх значень часового ряду, які будуть використані для прогнозування наступного значення.
- Параметр інтеграції ( $d$ ): вказує кількість разів, яку необхідно застосувати різниці до часового ряду щоб зробити його стаціонарним.
- Параметр ковзного середнього ( $q$ ): визначає кількість попередніх помилок прогнозу, які будуть використані для прогнозування наступного значення.

Вибір оптимальних значень параметрів ARIMA може бути виконаний з використанням різних методів, ручний вибір або ж автоматичний підбір через інформаційний критерій.

Правильне налаштування параметрів ARIMA відіграє важливу роль у якості прогнозу часового ряду. Неправильний вибір параметрів може призвести до недооцінки або переоцінки моделі, що може позначитися на точності прогнозів.

### 2.3.1 ACF та PACF

ACF (Autocorrelation Function) і PACF (Partial Autocorrelation Function) - це графіки, які використовуються для визначення параметрів моделі ARIMA [38].

ACF - це функція автокореляції, яка вимірює кореляцію між часовим рядом і його лагами. Лаг - це часова відстань між поточним значенням і попереднім значенням у ряді. ACF графік показує, як кожен лаг корелює з поточним значенням ряду. Якщо ACF для лага дорівнює 1, то це означає, що поточне значення ряду повністю залежить від значення на цьому лазі. ACF графік також дає змогу визначити наявність сезонності в даних.

PACF - це часткова функція автокореляції, яка вимірює кореляцію між часовим рядом і його лагами, виключаючи вплив інших лагів. PACF графік показує, як кожен лаг корелює з поточним значенням ряду, за умови, що всі інші лаги враховані. PACF графік дає змогу визначити, які лаги мають значущий внесок у поточне значення ряду.

Для визначення параметрів моделі ARIMA потрібен аналіз графіків ACF і PACF. Вибір оптимальних значень  $p$ ,  $d$  і  $q$  може бути складним завданням, особливо в разі, коли графіки ACF і PACF не дають чітких вказівок на значення параметрів. У таких випадках можуть використовуватися більш просунуті методи, як-от пошук за сіткою та алгоритми оптимізації.

### 2.3.2 Автоматичний підхід пошуку параметрів у сімействі моделі ARIMA

Хоча аналіз ACF і PACF може бути корисним для визначення параметрів моделі ARIMA, іноді він може бути складним і вимагати багато часу та зусиль. Крім того, наявність шуму в даних може ускладнити інтерпретацію графіків і ускладнити вибір параметрів. У зв'язку з цим нині все частіше використовують автоматичний підхід до визначення параметрів моделі ARIMA [39].

Один із таких підходів - це метод, відомий як автоматичне моделювання ARIMA (AutoARIMA), який використовує алгоритми оптимізації для вибору параметрів моделі ARIMA, що максимізують правдоподібність моделі та мінімізують помилку прогнозування. Цей метод автоматично оцінює початкові значення параметрів моделі ARIMA та визначає оптимальні значення параметрів, які можуть варіюватися залежно від типу даних та особливостей часового ряду.

Автоматичний підхід включає в себе використання алгоритмів, які автоматично знаходять найкращі параметри моделі ARIMA на основі мінімального значення інформаційного критерію, такого як AIC або BIC. Ці критерії оцінюють якість моделі та балансують між достатньою складністю моделі та її здатністю точно описувати дані.

В автоматичному підході можливо використовувати різні алгоритми, такі як ARIMA модель з автоматичним вибором параметрів (`auto.arima`) у програмі R, або функції ARIMA в пакетах Python `statsmodels` і `pmdarima`.

Крім того, в автоматичному підході ARIMA може використовувати інформацію про сезонність і сезонну різницю, щоб визначити параметри P, D і Q моделі SARIMA разом із параметрами  $p$ ,  $d$  і  $q$  для моделі ARIMA.

Auto ARIMA може бути дуже корисним інструментом для автоматичного вибору параметрів моделі ARIMA, особливо якщо часові ряди складні або мають багато сезонності.

### **2.3.3 Формалізація підходів до аналізу параметрів у сімействі моделі ARIMA**

Для визначення параметрів моделі ARIMA було представлено два підходи: ручний та автоматичний.

Ручний підхід ґрунтується на аналізі графіків ACF і PACF, які допомагають визначити параметри моделі. ACF графік показує залежність поточного значення часового ряду від його лагів, а PACF графік дає змогу визначити, які лаги роблять значущий внесок у поточне значення ряду. Однак, цей метод вимагає досвіду та експертної думки, і не завжди дає змогу точно визначити оптимальні параметри моделі.

Автоматичний підхід ґрунтується на алгоритмах оптимізації, які шукають оптимальні значення параметрів моделі ARIMA. У цьому випадку, не потрібно аналізувати графіки ACF і PACF, що робить цей підхід більш швидким і автоматичним. Однак, цей метод може бути менш точним і не завжди дає оптимальні параметри.

Обидва підходи мають свої переваги та недоліки, і вибір методу залежить від конкретного завдання та уподобань дослідника.

## **2.4 Система автоматизованого прогнозування часових рядів**

Створення прогнозів часових рядів є нетривіальним завданням, яке потребує глибоких знань у галузі статистики та математичного моделювання. Необхідно враховувати велику кількість чинників, які можуть вплинути на досліджуваний процес або явище, і визначити їхній вплив на майбутнє значення часового ряду. Це вимагає від аналітика великого обсягу роботи, яка може бути дуже трудомісткою і витратною.

У зв'язку з цим виникає необхідність у створенні системи автоматизованого прогнозування часових рядів, що дасть змогу скоротити час і зусилля, що витрачаються на побудову прогнозів. Така система може містити в собі попередню обробку даних, автоматичний вибір найбільш підходящих

методів прогнозування, а також візуалізацію отриманих результатів. Завдяки такій системі, користувач може отримати прогнози часових рядів з мінімальною участю та автоматизувати процес прогнозування. Створення програми, яка є реалізацією такої системи, є актуальним завданням.

Основною метою створення системи автоматизованого прогнозування часових рядів є допомога аналітикам у їх роботі та збільшенні швидкості ухвалення управлінських рішень у бізнесі. Швидке і точне прогнозування дасть змогу передбачити зміни ринку, поведінку споживачів, а також допоможе оптимізувати виробничі процеси та управління фінансами компанії.

Також система автоматизованого прогнозування часових рядів може бути корисною для запобігання втрат у бізнесі. Наприклад, прогнозування попиту на товари та послуги дасть змогу скоротити втрати, пов'язані з перевиробництвом або недовиробництвом продукції. Автоматизоване прогнозування може також допомогти в управлінні ризиками, пов'язаними з фінансами та інвестиціями, що є важливим аспектом у сучасному бізнесі.

Така система повинна містити у собі всі основні етапи побудови прогнозів, починаючи від завантаження даних, до вибору та навчання моделей, виконання та візуалізації прогнозування.

Таким чином, створення системи автоматизованого прогнозування часових рядів має велике значення для бізнесу, оскільки дає змогу скоротити час і витрати на побудову прогнозів, збільшити точність і швидкість ухвалення управлінських рішень, а також знизити ризики втрат і поліпшити управління фінансами та виробничими процесами.

## **2.5 Концепція інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів**

Інформаційне забезпечення - це система, спрямована на подання користувачеві інформації. Вона обробляє вхідні дані від користувача, аналізує та надає результати аналізу користувачу для прийняття обґрунтованих

управлінських рішень. Воно охоплює різні технології, методи та інструменти, які дають змогу обробляти та подавати інформацію в зручній і зрозумілій формі для користувача. Розроблена система є інформаційним забезпеченням з аналізу часових рядів, що дає змогу користувачам задавати свої часові ряди й одержувати відповідну інформацію щодо них. Вона забезпечує можливість вибору різних методів прогнозування і надає візуалізацію прогнозу, що дає змогу користувачам отримувати наочне уявлення про майбутні тренди і патерни в їхніх часових рядах. Це дає змогу користувачам ухвалювати більш усвідомлені рішення на основі аналізу та прогнозування часових рядів. Система автоматизованого прогнозування часових рядів - це система інформаційного забезпечення, що містить у собі автоматизацію процесу для зменшення безпосередньої участі користувача в процесі отримання інформації.

Для створення інформаційного забезпечення та системи автоматизованого прогнозування необхідно використовувати такі інформаційні та програмні технології:

- Мови програмування: Для розробки інформаційного забезпечення та систем автоматизованого програмування можуть бути використані мови програмування, такі як Python, Java, C++, R та інші. Ці мови пропонують потужні інструменти для опрацювання даних, реалізації алгоритмів і розроблення користувацьких інтерфейсів.
- Бібліотеки для роботи з часовими рядами: Важливо вибрати відповідні бібліотеки, які надають функції та інструменти для маніпулювання, аналізу та візуалізації часових рядів. Деякі популярні бібліотеки для роботи з часовими рядами в Python включають Pandas, NumPy, Matplotlib, ARIMA, SARIMA та інші.
- Інтерактивні інструменти та візуалізація: Для зручної роботи з часовими рядами рекомендується використовувати інтерактивні інструменти та візуалізацію. Jupyter Notebook або JupyterLab надають інтерактивне

середовище для аналізу даних і створення звітів з кодом, текстом і графіками.

- Середовище розробки: Рекомендується використовувати середовище розробки, яке забезпечує зручну роботу з кодом, налагодження та управління проектом. Однією з популярних середовищ розробки для Python є PyCharm, яка пропонує широкий набір функцій та інструментів для розробки та налагодження програм.

Важливо враховувати особливості завдання і вимоги проекту під час вибору інформаційних і програмних технологій. Комбінація Python, бібліотек для роботи з часовими рядами, PyCharm як середовища розроблення та інтерактивних інструментів, як-от Jupyter Notebook, забезпечить ефективне розроблення та реалізацію інформаційного забезпечення. Важливо зазначити, що мова програмування R та її відповідні бібліотеки, також надають велику можливість в аналізі даних [40]. Також розглянута система підтримки ухвалення рішень із прогнозуванням [25] може розглядатися однією з варіацій інформаційного забезпечення.

### **2.5.1 Інструменти для інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів**

Python - це високорівнева, інтерпретована мова програмування загального призначення, яку було створено Гвідо ван Россумом. Мова Python швидко стала популярною в різних галузях, включно з науковими та дослідницькими роботами, веб-розробкою, аналізом даних, машинним навчанням і багатьом іншим.

Python має простий і зрозумілий синтаксис, що робить його дуже доступним для новачків. Завдяки цьому, Python став однією з найпопулярніших мов програмування, особливо в галузі аналізу даних і машинного навчання.

Python має велику й активну спільноту, яка створює бібліотеки та фреймворки для вирішення різних завдань.

Python має безліч бібліотек для роботи з науковими обчисленнями та аналізом даних, таких як NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow і багато інших. У сукупності ці бібліотеки надають великий набір інструментів для роботи з даними, навчання моделей машинного навчання, візуалізації та багато іншого.

Python має відкритий вихідний код і безкоштовний для використання, що робить його доступним для будь-якого користувача. Він також підтримує різні операційні системи, зокрема Windows, Mac OS і Linux. Загалом, Python є потужним і гнучким інструментом для програмування та розв'язання різноманітних завдань, включно з аналізом і прогнозуванням часових рядів.

Бібліотеки NumPy, Pandas і Matplotlib є одними з найбільш популярних і потужних інструментів в екосистемі Python для аналізу даних. Ось короткий опис кожної з цих бібліотек:

NumPy (Numerical Python) надає потужні структури даних, такі як багатовимірні масиви (ndarray), і набір функцій для роботи з числовими даними. Він має високу продуктивність і ефективність, що робить його ідеальним для виконання математичних і наукових обчислень. NumPy надає функціональність для виконання операцій над масивами, лінійної алгебри, випадкових чисел, перетворень Фур'є і багато чого іншого.

Pandas - це бібліотека, побудована поверх NumPy, яка надає зручні та ефективні структури даних та інструменти для аналізу й маніпулювання табличними даними. Основними структурами даних у Pandas є об'єкти Series (одновимірні дані) і DataFrame (двовимірні дані). Pandas надає потужні функції для фільтрації, сортування, агрегування, злиття і перетворення даних.

Matplotlib - це бібліотека візуалізації даних, яка дає змогу створювати високоякісні графіки, діаграми та візуальні подання. Вона надає широкий набір функцій для створення різних типів графіків, включно з лінійними графіками, стовпчиковими діаграмами, точковими діаграмами, гістограмами, круговими

діаграмами та багато іншого. Matplotlib має гнучкість і налаштованість, що дає змогу користувачам створювати графіки з високим рівнем деталізації та контролю.

Разом ці бібліотеки надають потужний інструментарій для аналізу, моделювання та візуалізації даних, зокрема часових рядів у Python. Їх широко застосовують у наукових дослідженнях, аналітичних завданнях і бізнес-додатках, де потрібне опрацювання та аналіз даних.

Jupyter Notebook - це інтерактивне середовище програмування, що дає змогу створювати та виконувати код, візуалізувати дані, а також створювати документи, які об'єднують код, текст і мультимедійні елементи в єдиний документ. Він використовується для різних завдань, включно з науковими дослідженнями, обробкою даних, машинним навчанням і візуалізацією даних.

Jupyter Notebook підтримує безліч мов програмування, зокрема Python, R, Julia та інші. Він дає змогу писати і налагоджувати код покроково, а також створювати графічні елементи, включно з діаграмами, графіками і таблицями.

Однією з особливостей Jupyter Notebook є можливість створювати документи, що об'єднують код, текст і графічні елементи в одному місці. Це робить його зручним інструментом для обміну знаннями та спільної роботи над проектами.

Jupyter Notebook доступний як локальний застосунок, а також через інтернет, що дає змогу користувачам працювати з даними та кодом з будь-якого місця. Він широко використовується в наукових дослідженнях, обробці даних і машинному навчанні, а також в освіті.

PyCharm - це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. Воно надає різні інструменти та функції, що спрощують процес розробки, налагодження та управління проектами мовою Python.

PyCharm пропонує безліч можливостей, які допомагають розробникам підвищити продуктивність і ефективність у роботі. Ось деякі з ключових особливостей PyCharm:

1. Редактор коду: PyCharm має потужний і зручний редактор коду, який пропонує підсвічування синтаксису, автодоповнення, перевірку помилок, рефакторинг та інші функції для поліпшення процесу написання коду.
2. Налаштування: Інструменти налаштування в PyCharm дають змогу розробникам виконувати покрокове налаштування коду, встановлювати точки зупинки, аналізувати значення змінних і шукати помилки в програмі.
3. Керування проектом: PyCharm надає функціональність для керування проектами, включно з системою контролю версій, інтеграцією з Git.
4. Аналіз коду: PyCharm здійснює статичний аналіз коду, виявляючи потенційні помилки, пропонуючи підказки щодо поліпшення коду і перевіряючи відповідність стандартам оформлення коду.
5. Інтеграція з іншими інструментами: PyCharm інтегрується з різними зовнішніми інструментами та сервісами, як-от бази даних, веб-фреймворки, системи складання та багато іншого.

PyCharm доступний у двох версіях: Community Edition (безкоштовна версія) і Professional Edition (платна версія з усіма розширеними функціями). Обидві версії надають потужні можливості розробки та широкі можливості для роботи з проектами мовою Python.

### **2.5.2 Концептуальна модель інформаційного забезпечення прийняття управлінських рішень методами аналізу часових рядів**

Початково розробка була проведена в середовищі Jupyter Notebook, де було створено інформаційне забезпечення для аналізу та моделювання часових рядів. У Jupyter Notebook мали можливість вручну змінювати вхідні дані, налаштовувати різноманітні функції, окремо виводити графіки та виконувати

інші дії. Це давало змогу гнучко керувати процесом аналізу та моделювання часових рядів, а також проводити інтерактивне дослідження даних. У такому інформаційному забезпеченні на Jupyter Notebook було реалізовано всі основні етапи процесу побудови прогнозів. Розглянемо більш детально кожен із цих етапів:

- Завантаження даних: Система надає можливість завантаження даних часового ряду, таких як файли типу csv. Користувач може вказати шлях до файлу.
- Попередній аналіз даних: Система проводить попередній аналіз завантажених даних, виставлення формату дати в часовому ряді, що включає в себе перевірку на наявність пропусків, викидів. Також виконується декомпозиція часового ряду.
- Візуалізація даних: Система дає змогу візуалізувати завантажені часові ряди за допомогою графіків. Це допомагає користувачеві візуально оцінити тенденції, цикли, сезонність та інші особливості даних.
- Налаштування параметрів: Користувач має можливість налаштовувати різні параметри моделей і методів прогнозування, щоб досягти найкращих результатів. Це може включати вибір вікна прогнозування, визначення оптимальних значень параметрів моделей або використання різних стратегій адаптації моделей до мінливих умов.
- Підбір моделі: Система надає набір стандартних моделей прогнозування часових рядів, таких як ARIMA, експоненціальне згладжування та інші. Користувач може вибрати потрібну модель або використовувати автоматичний підбір моделі на основі середньої абсолютної помилки у відсотках (MAPE).
- Прогнозування: Система дає змогу виконувати прогнозування майбутніх значень часового ряду на основі обраної моделі. Користувач може вказати горизонт прогнозу - період часу, на який

потрібно отримати прогнозні значення. Система генерує прогнози і виводить їх користувачеві в зручній формі, наприклад, у вигляді числових значень або графіка.

У Jupyter Notebook кожен з вищевказаних етапів побудови прогнозів часових рядів було реалізовано у вигляді окремих блоків коду, які можна було запустити і керувати вручну. У цього підходу є свої переваги:

- **Інтерактивність:** Jupyter Notebook надає інтерактивне середовище, де користувач може взаємодіяти з кодом, змінювати параметри моделей, візуалізувати результати і спостерігати зміни в реальному часі. Це дає змогу гнучкіше і швидше налаштовувати та перевіряти різні варіанти моделей і методів.
- **Графічні можливості:** Jupyter Notebook підтримує вбудовану візуалізацію даних і результатів, що полегшує аналіз часових рядів і дає змогу легко представити результати в графічному вигляді. Графіки можуть бути налаштовані та відображені поруч із кодом, що спрощує візуальне вивчення та аналіз даних.
- **Гнучкість і модульність:** Блоки коду в Jupyter Notebook можна виконувати незалежно один від одного, що дає змогу користувачеві змінювати порядок виконання етапів, повторювати ітерації та модифікувати код у міру необхідності. Це робить процес моделювання більш гнучким і модульним, даючи змогу швидко вносити зміни й адаптуватися до нових вимог.

Але цей підхід не враховує автоматизованого підходу. При роботі з Jupyter Notebook вручну необхідно запускати кожен блок коду по черзі та керувати процесом моделювання. Це вимагає постійного втручання користувача і може бути неефективним і вимагати повторного виконання одних і тих самих операцій. Також введення значень, які користувач хоче застосувати до аналізу часового ряду, наприклад на скільки значень вперед він хоче отримати прогноз,

є не зовсім зручним, такі значення треба мати десь в окремому місці, а також може знадобитися необхідність вручну змінювати значення всередині коду.

Тому для розв'язання проблеми відсутності автоматизації в процесі прогнозування часових рядів, було розроблено окремий консольний додаток, спеціалізований на автоматизації цього процесу. Цей застосунок було створено, як уже було згадано, мовою Python з використанням інтегрованого середовища розробки PyCharm.

На відміну від Jupyter Notebook, де користувачеві потрібно було вручну керувати блоками коду та змінювати параметри моделей, цей автоматизований застосунок надає користувачеві готове рішення, де прогнозування часових рядів виконується автоматизовано з мінімальною участю користувача.

У нього було перенесено основний функціонал із попередньої системи, як-от перевірка та обробка пропущених значень, виділення аномалій, автоматичний вибір найкращої моделі, також просто вибір окремих моделей, і виведення результатів. Також було додано функцію збереження результатів прогнозування в робочу папку, яку користувач повинен вказати на початку роботи додатка.

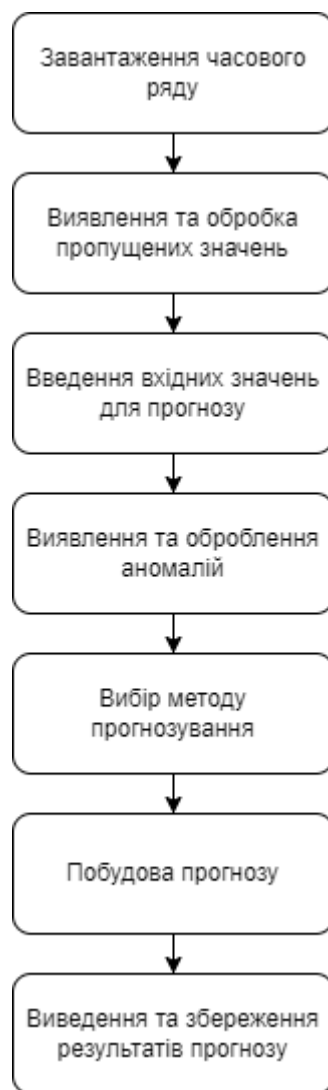


Рисунок 2.1 - Схема роботи програми

Таким чином, цей окремий застосунок дає змогу користувачам отримати прогнози часових рядів із мінімальними зусиллями та взаємодією. Користувач може завантажити дані, ввести ввідні значення, обрати метод прогнозування, який забажає, зокрема можна обрати автоматичний підбір моделі і в підсумку отримати результат прогнозування без необхідності вручну налаштовувати параметри та виконувати кроки моделювання. Це значно підвищує ефективність і зручність процесу прогнозування часових рядів. Така система може бути зручним інструментом для спеціалістів у галузі економіки, фінансів, маркетингу та інших сферах, де аналіз і прогнозування часових рядів відіграють важливу роль.

Тому в розділі моделювання буде розглянуто систему автоматизованого прогнозування часових рядів, яка вже містить у собі основний функціонал. Буде розглянуто програмний код такої системи, а також опис який функціонал забезпечує розглянутий код.

## **2.6 Висновок по другому розділу**

У цьому розділі формалізовано інформацію щодо застосування часових рядів для прогнозування. Встановлено, що часові ряди являють собою послідовність спостережень, упорядкованих у часі, і широко застосовуються в різних галузях, як-от фінанси, економіка, маркетинг, операційне управління та прогнозування.

Вивчено різні методи прогнозування часових рядів, включно з такими сімействами методів прогнозування як експоненціальне згладжування та ARIMA. Кожен із цих методів має свої переваги та обмеження, і вибір відповідного методу залежить від характеристик часового ряду та поставлених завдань.

Розглянуто актуальність в автоматизації та спрощення процесу прогнозування часових рядів, зокрема створення системи автоматизованого прогнозування часових рядів. Проведено концептуалізацію такого інформаційного забезпечення. Ця система містить у собі всі основні етапи побудови прогнозів, починаючи від завантаження даних, до вибору та навчання моделей, виконання та візуалізації прогнозування. Завдяки розробленому застосунку, користувач може отримати прогнози часових рядів з мінімальною участю та автоматизувати процес прогнозування.

Для створення інформаційного забезпечення та системи автоматизованого програмування було вивчено та вирішено використати такі інформаційні та програмні технології як мова програмування Python з її бібліотеками для аналізу даних, інтерактивний інструмент Jupyter Notebook, а також інтегроване середовище розробки PyCharm.

Отже, вивчення часових рядів і методів їхнього прогнозування являє собою важливу галузь аналізу даних, що має практичне застосування в бізнесі. Автоматизовані системи прогнозування дають змогу ефективно використовувати час і ресурси та ухвалювати більш обґрунтовані рішення на основі аналізу часових рядів.

## **РОЗДІЛ 3. МОДЕЛЮВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ**

### **3.1 Вступ до розділу**

У цьому розділі буде розглянуто розробку автоматизованої системи прогнозування часових рядів у вигляді консольного додатка, написаного мовою програмування Python і його відповідних бібліотек для аналізу та прогнозування часових рядів, з використанням інтегрованого середовища розробки PyCharm.

### **3.2 Система автоматизованого прогнозування часових рядів**

Використано мову програмування Python та її бібліотеки, такі як Pandas, Matplotlib і інші, для завантаження даних, візуалізації часових рядів, а також для створення прогнозних моделей. Також розглянуто різні методи аналізу часових рядів, як-от STL декомпозиція, Exponential smoothing і ARIMA моделі, і застосовано їх для створення прогнозних моделей на основі історичних даних. Після цього можливо використовувати ці моделі для прогнозування майбутніх значень часових рядів, що дає змогу ухвалювати обґрунтовані управлінські рішення.

Насамперед під час роботи в PyCharm, як і в будь-якому середовищі програмування, необхідно завантажити необхідні бібліотеки. Бібліотеки - це набори функцій та інструментів, які програмісти використовують для спрощення роботи та розв'язання завдань.

У нашому випадку, для аналізу часових рядів, буде використано такі бібліотеки, як pandas, numpy, matplotlib, statsmodels тощо. Кожна з них надає набір інструментів для роботи з даними, і без їх використання аналіз може бути неповним або навіть неможливим.

Завантаження додаткових бібліотек у Python здійснюється за допомогою команди "import". Наприклад, команда "import pandas as pd" завантажує

бібліотеку pandas і дає змогу звертатися до її функцій і методів через псевдонім "pd". Це спрощує написання коду і робить його більш читабельним.

Бібліотеки, які завантажуюмо, представлені нижче.

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot
import pmdarima as pm
from datetime import datetime
from datetime import timedelta
from pandas.plotting import register_matplotlib_converters
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.statespace.sarimax import SARIMAX
register_matplotlib_converters()
from time import time
from matplotlib import pyplot as plt, dates as mdates
from statsmodels.tsa.seasonal import STL
from statsmodels.tsa.api import ExponentialSmoothing
```

Цей код є блоком імпорту необхідних бібліотек і модулів для аналізу та моделювання часових рядів у Python.

- os - модуль для роботи з операційною системою;
- pandas - бібліотека для роботи з даними, заснована на DataFrame;
- numpy - бібліотека для наукових обчислень у Python;
- matplotlib.pyplot - бібліотека для візуалізації даних;
- pmdarima - бібліотека для автоматичного підбору параметрів моделі ARIMA;
- datetime і timedelta - модулі для роботи з датами і часом;

- `acf` і `pacf` - функції для оцінювання автокореляційної та приватної автокореляційної функцій часових рядів;
- `SARIMAX` - клас для створення моделей SARIMA (сезонна ARIMA);
- `register_matplotlib_converters` - функція, що реєструє конвертери для `matplotlib`;
- `time` - модуль для роботи з часом;
- `pyplot` і `dates` з `matplotlib` - модулі для візуалізації даних;
- `STL` - клас для сезонної декомпозиції часових рядів;
- `ExponentialSmoothing` - клас для побудови експоненціального згладжування.

Імпорт необхідних бібліотек і модулів є необхідним кроком під час початку роботи з часовими рядами в Python, оскільки ці інструменти надають функціонал для опрацювання, аналізу та моделювання часових рядів.

```
working_directory = input("Введіть робочу папку: ")
os.chdir(working_directory)
print("WD: ", os.getcwd())
```

Цей код запитує у користувача введення робочої директорії (`working directory`), потім встановлює її як поточну робочу директорію за допомогою функції `os.chdir()`. Потім він виводить поточну робочу директорію за допомогою функції `os.getcwd()`.

Наприклад, користувач може ввести `C:/Users/Username/Documents`, що означає, що робоча директорія тепер буде `C:/Users/Username/Documents`.

Цей код важливий, тому що він дає змогу визначити робочу директорію для цього проекту, що спрощує завантаження і збереження файлів і забезпечує чистіший та організованіший процес роботи з даними і моделями.

Після завантаження додаткових бібліотек, та встановлення робочої папки переходимо до завантаження даних. У нашому випадку, потрібно завантажити часовий ряд, який може являти собою дані про продажі, клієнтів, ціни на товари тощо. Запитуємо у користувача повну назву csv файлу з часовим рядом. Важливо, щоб файл часового ряду мав певний формат, де перший стовпчик містить дати, а другий стовпчик містить значення.

```
file_name = input("Введіть назву файлу з даними: ")  
ts = pd.read_csv(file_name)
```

Після завантаження файлу можемо використовувати функцію `pd.read_csv()` з бібліотеки `pandas` для читання даних і завантаження їх в об'єкт `DataFrame`.

```
ts.iloc[:,0] = pd.to_datetime(ts.iloc[:,0]).dt.strftime('%Y-%m-%d')
```

Потім можемо використовувати методи `pandas` для обробки даних. Наприклад, можемо використовувати метод `pd.to_datetime()` для перетворення стовпця з датами у формат `datetime`. Потім використовує метод `strftime()` для перетворення дати і часу у формат рік-місяць-день. Результатом буде новий стовпчик у форматі дати і часу, який буде використовуватися для подальшого аналізу і прогнозування часового ряду.

```
ts.set_index(ts.iloc[:,0],inplace = True)  
ts.drop(ts.columns[0], axis=1, inplace=True)
```

Потім як індекс ряду встановлюється стовпець із датою і часом, використовуючи метод `set_index()`, а перший стовпець, який тепер є дублюючим індексом, видаляється за допомогою методу `drop()`.

У результаті маємо часовий ряд у форматі `DataFrame`, готовий для аналізу.

Наступний код стосується попередньої обробки (preprocessing) даних часового ряду.

```
nul_data = pd.isnull(ts.iloc[:,0])
if not ts[nul_data].empty:
    print("У часовому ряду виявлено пропущені значення. Яким способом ви хочете їх виправити?")
    print("Доступні такі варіанти як: mean, median, locf, nocb, linear, spline, delete.")
    preparation = input("Напишіть, який варіант хочете використати: ")
    if preparation == 'delete':
        ts = ts.dropna(axis = 0, how = 'any')
    if preparation == 'mean':
        ts.iloc[:,0] = ts.assign(FillMean=ts.iloc[:,0].fillna(ts.iloc[:,0].mean())).iloc[:,1]
    if preparation == 'median':
        ts.iloc[:,0] = ts.assign(FillMean=ts.iloc[:,0].fillna(ts.iloc[:,0].median())).iloc[:,1]
    if preparation == 'locf':
        ts.iloc[:,0] = ts.iloc[:,0].fillna(method = 'ffill')
    if preparation == 'nocb':
        ts.iloc[:,0] = ts.iloc[:,0].fillna(method = 'bfill')
    if preparation == 'linear':
        ts.iloc[:,0] = ts.iloc[:,0].interpolate(method='linear')
    if preparation == 'spline':
        ts.iloc[:,0] = ts.iloc[:,0].interpolate(option='spline')
```

Далі створюється змінна `nul_data`, яка містить булевий масив, що вказує на місця, де в ряді знаходяться пропущені значення.

Потім виконується перевірка, чи містить часовий ряд пропущені значення. Якщо пропущені значення виявлено, користувач повинен вибрати, який метод використовувати для їх заповнення

У цьому коді представлено такі методи заповнення пропущених значень:

- "delete" - видалення рядків із пропущеними значеннями.
- "mean" - заповнення пропущених значень середнім значенням по ряду.
- "median" - заповнення пропущених значень медіанним значенням по ряду.
- "locf" - заповнення пропущених значень попередніми (last observation carried forward).
- "nosc" - заповнення пропущених значень наступними (next observation carried backward).
- "linear" - заповнення пропущених значень лінійною інтерполяцією.
- "spline" - заповнення пропущених значень інтерполяцією методом сплайнів.

Для обраного методу заповнення пропущених значень, використовується відповідний метод `pandas`, який застосовується до першого стовпця часового ряду (який містить значення). Заповнений часовий ряд зберігається назад у змінну `ts`, щоб була можливість використовувати його далі в нашому аналізі.

```
horizon = int(input("На скільки значень вперед ви хочете побудувати прогноз? "))
season_period = int(input("Яка кількість сезонних періодів у часовому ряду?
(введіть значення 1 якщо сезонності немає) "))
```

У цій ділянці коду користувачеві пропонується ввести два параметри, необхідні для побудови прогнозу часового ряду.

Перший параметр `horizon` - це кількість майбутніх значень, які хочемо передбачити. Наприклад, якщо працюємо зі щоденними даними і хочемо побудувати прогноз на два тижні вперед, то значення `horizon` має дорівнювати 14.

Другий параметр `season_period` - це кількість часових одиниць в одному сезонному періоді. Наприклад, якщо працюємо зі щоденними даними і помічаємо, що в даних є явна сезонність із періодом у 7 днів, то значення `season_period` має дорівнювати 7. Якщо сезонності в даних немає, то значення `season_period` має дорівнювати 1.

Наступний код перевіряє наявність сезонності в часовому ряді та знаходить можливі викиди в ряді за допомогою STL-декомпозиції. Якщо в ряді є сезонність, то метод STL розділяє ряд на трендову, сезонну і залишкову складові. Потім для залишкової складової обчислюються середнє і стандартне відхилення, і будуються (за правилами 3 сигм) верхній і нижній пороги для визначення викидів (аномалій). Якщо в ряді є викиди, то пропонується замінити їх на більш відповідні значення.

```
if season_period != 1:
    stl = STL(ts.iloc[:,0], period = season_period)
    result = stl.fit()
    resid = result.resid

    resid_mean = resid.mean()
    resid_std = resid.std()

    upper = resid_mean + 3*resid_std
    lower = resid_mean - 3*resid_std

    anomalies = ts[(resid < lower) | (resid > upper)]
    anomalies = anomalies.reset_index()
    outer = anomalies.merge(ts[nul_data], how='outer', indicator = True)
    anomalies = outer[(outer._merge=='left_only')].drop('_merge', axis=1)
    anomalies.set_index(anomalies.iloc[:,0], inplace = True)
```

```
anomalies.drop(anomalies.columns[0], axis=1, inplace=True)
print(anomalies)
```

```
if not anomalies.empty:
```

```
    anomquest = input("Виявлено можливі викиди в часовому ряду. Замінити  
їх на більш відповідні значення? (Так/Ні) ")
```

```
    if anomquest == "Так" or anomquest == "так":
```

```
        i = 0
```

```
        while i < len(anomalies.index):
```

```
            ts[ts.index == anomalies.index[i]] = np.nan
```

```
            i = i + 1
```

```
        ts.iloc[:,0] = ts.iloc[:,0].interpolate(method='linear')
```

Більш докладно:

- if season\_period != 1: - якщо в ряді є сезонність
- stl = STL(ts.iloc[:,0], period = season\_period) - виконується STL-декомпозиція ряду із заданим сезонним періодом
- resid = result.resid - отримуємо залишкову складову з декомпозиції
- resid\_mean = resid.mean(), resid\_std = resid.std() - обчислюємо середнє і стандартне відхилення залишків
- upper = resid\_mean + 3\*resid\_std, lower = resid\_mean - 3\*resid\_std - визначаємо (за правилами 3 сигм) верхній і нижній пороги для викидів (аномалій)
- anomalies = ts[(resid < lower) | (resid > upper)] - знаходимо викиди в ряді
- outer = anomalies.merge(ts[nul\_data], how='outer', indicator = True) - об'єднуємо викиди з пропущеними значеннями
- anomalies = outer[(outer.\_merge=='left\_only')].drop('\_merge', axis=1) - залишаємо (через використання anti-join) тільки ті викиди, які не були початково пропущеними значеннями

- `anomalies.set_index(anomalies.iloc[:,0],inplace=True), anomalies.drop(anomalies.columns[0], axis=1, inplace=True)` - встановлюємо дати викидів як індекс, видаляємо перший стовпець
- `print(anomalies)` - виводимо аномалії
- `anomquest = input("Виявлено можливі викиди в часовому ряду. Замінити їх на більш відповідні значення? (Так/Ні) ")` - ставимо запитання, чи потрібно замінити викиди
- `ts[ts.index == anomalies.index[i]] = np.nan, ts.iloc[:,0] = ts.iloc[:,0].interpolate(method='linear')` - якщо відповідь була так, то замінюємо викиди на пропущені значення та інтерполюємо пропущені значення лінійно.

Оскільки було згадано про правило 3 сигм, а також про anti-join, то розповімо про це детальніше.

Правило 3-х сигм [41], також відоме як правило трьох сигм або правило 68-95-99.7, є статистичним правилом, що використовується для вимірювання ймовірнісної значущості результатів експерименту. Це правило говорить, що за нормального розподілу даних, близько 68% усіх спостережень мають перебувати в межах одного стандартного відхилення від середнього значення, близько 95% - у межах двох стандартних відхилень, а близько 99.7% - у межах трьох стандартних відхилень.

Це правило широко використовується в багатьох галузях, таких як фізика, хімія, біологія, економіка тощо. Наприклад, якщо є вибірка зі 100 значень, які розподілені нормально із середнім значенням 50 і стандартним відхиленням 5, то 68% із цих значень лежатимуть у діапазоні від 45 до 55, 95% - у діапазоні від 40 до 60, а 99.7% - у діапазоні від 35 до 65.

Правило 3-х сигм може використовуватися для виявлення викидів у даних. Якщо спостереження перебуває за межами трьох стандартних відхилень від середнього значення, то воно може вважатися викидом і потенційно виключене з подальшого аналізу.

Anti-join - це операція з'єднання таблиць, за якої в результуючій таблиці залишаються тільки ті рядки, які не мають відповідних значень в іншій таблиці. Інакше кажучи, anti-join дає змогу відфільтрувати з однієї таблиці ті рядки, які не мають відповідних значень в іншій таблиці, тобто є такими, що "не збігаються".

Anti-join часто використовується в SQL для виконання операцій фільтрації даних. Наприклад, якщо у нас є дві таблиці - таблиця замовлень і таблиця клієнтів, і є завдання вибрати тільки ті замовлення, які не належать клієнтам певного міста, то є можливість виконати anti-join, щоб виключити замовлення, пов'язані з цим містом.

У Pandas anti-join можна виконати за допомогою методу merge(), вказавши параметр how='outer' і явно вказавши стовпці для з'єднання [42]. Потім можна вибрати тільки ті рядки, які присутні тільки в лівій таблиці (тобто виконали anti-join).

```
print("Який метод аналізу часових рядів ви хочете застосувати?")
print("Доступні такі варіанти як:")
print("1) ExponentialSmoothing;")
print("2) ARIMA;")
print("3) Hybrid (середнє значення двох попередніх методів);")
print("4) Автоматичний вибір;")
print("5) Вивести всі три методи.")
choice = int(input("Введіть номер варіанта, який хочете застосувати: "))
```

Цей код запитує користувача, який метод аналізу часових рядів він хоче застосувати, та пропонує на вибір кілька варіантів. Користувач має вибрати номер варіанту і ввести його у консоль. Варіанти, які пропонуються користувачеві, є наступні:

1. ExponentialSmoothing;
2. ARIMA;

3. Hybrid (середнє значення двох попередніх методів);
4. Автоматичний вибір;
5. Вивести всі три методи.

Залежно від вибору користувача, буде виконано певний метод аналізу часових рядів.

```
sestype = None
train_length = len(ts) - horizon
ts_train = ts.iloc[0:train_length]
ts_test = ts.iloc[train_length:len(ts)]
```

У цьому коді визначаються кілька змінних для подальшого використання в аналізі часових рядів:

- `sestype` - змінна, яка використовується для зберігання типу сезонності (якщо вона є) в даних часового ряду (необхідна змінна для експоненціального згладжування). Спочатку їй присвоєно значення `None`.
- `train_length` - змінна, яка зберігає довжину навчальної вибірки. Довжина навчальної вибірки визначається як загальна кількість точок даних у часовому ряду за вирахуванням горизонту передбачення (кількість точок, які необхідно передбачити).
- `ts_train` - змінна, яка зберігає навчальну вибірку, що містить усі точки даних, крім останніх `horizon` точок. Для її визначення використовується метод `.iloc` для вилучення підмножини даних з вихідного часового ряду `ts`.
- `ts_test` - змінна, яка зберігає тестову вибірку, що містить останні `horizon` точок даних часового ряду. Для її визначення також використовується метод `.iloc` для вилучення підмножини даних з вихідного часового ряду `ts`.

Тестова та навчальна вибірка необхідна буде, якщо користувач вибере автоматичний метод, тоді за навчальною вибіркою моделі будуватимуть прогнози, а вони вже порівнюватимуться з тестовими значеннями. Усе це дасть

змогу вибрати модель із меншою помилкою в передбаченні, і її вже буде обрано як основну модель для побудови прогнозу на всю довжину часового ряду.

Нижче представлено дві функції, які використовуються для побудови моделі експоненціального згладжування на навчальній вибірці, де в підсумку виводяться її значення MAPE.

```
def exptype(x, y):
    if season_period == 1:
        model = ExponentialSmoothing(x, seasonal_periods=None, trend=
'additive', seasonal=y, damped_trend = True).fit()
    else:
        model = ExponentialSmoothing(x, seasonal_periods=season_period,
trend= 'additive', seasonal=y, damped_trend = True).fit()
    return model, model.aic
```

```
def exptest(horizon):
    modeladd = exptype(ts_train, 'additive')
    modelmul = exptype(ts_train, 'multiplicative')
    if modeladd[1] < modelmul[1]:
        model = modeladd[0]
        sestype = 'additive'
    else:
        model = modelmul[0]
        sestype = 'multiplicative'

    fitted = model.predict(0, len(ts_train) + horizon - 1)
    forecastexp = fitted.iloc[len(ts_train):len(fitted)]
    forecastexp1 = forecastexp.to_numpy()
    ts_test1 = ts_test.to_numpy()
```

```
return (np.mean(np.abs((forecastexp1 - ts_test1)/ts_test1))*100), forecastexp1,
sestype
```

Цей код визначає функцію `exptype(x, y)`, яка приймає два параметри  $x$  і  $y$ . Змінна `season_period` використовується для перевірки наявності сезонності в даних.

Якщо сезонність є (тобто `season_period` не дорівнює 1), то модель експоненціального згладжування (`ExponentialSmoothing()`) створюється з використанням зазначеної сезонності `season_period`. Якщо ж сезонності немає, то модель створюється з `seasonal_periods=None`.

Під значенням `u` передається тип сезонності. В обох випадках використовується адитивний тренд і затухаючий тренд (`damped_trend = True`).

Функція `exptype()` повертає модель та її інформаційний критерій Акаїке.

Функція `exptest(horizon)` використовує `exptype()` для побудови моделі на даних тренувальної вибірки (`ts_train`). Вона будує дві моделі (одна з адитивною, а інша з мультиплікативною сезонністю) з використанням `exptype()` і обирає ту, у якої менший АІС. А який тип сезонності у цієї моделі записується у змінну `sestype`, що дасть змогу надалі не переобчислювати яку сезонність використовувати вже на початкових, загальних даних.

Потім модель використовується для створення прогнозу на `horizon` кроків уперед. Прогноз зберігається у змінній `forecastexp1`.

Функція `exptest()` повертає середнє абсолютне значення процентної помилки прогнозу `MAPE`, прогнози `forecastexp1` і тип сезонності `sestype`. Ці значення знадобляться для вибору кращої моделі, а також побудови гібридної моделі.

Нижче представлено функцію, яка вже використовується для побудови моделі `ARIMA` на навчальній вибірці, де в підсумку виводяться її значення `MAPE`.

```
def armtest(horizon, season_period):
```

```

model = pm.auto_arima(ts_train, start_p=2, d=None, start_q=2, max_p=5,
                      max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2,
                      max_D=1, max_Q=2, max_order=5, m=season_period)
model.fit(ts_train)
forecastarm = model.predict(n_periods=horizon)
forecastarm1 = forecastarm.to_numpy().flatten()
ts_test1 = ts_test.to_numpy()
return (np.mean(np.abs((forecastarm1 - ts_test1)/ts_test1))*100), forecastarm1

```

Цей код реалізує функцію `armtest`, яка використовує модуль `pm` для автоматичного підбору оптимальних параметрів моделі ARIMA. Функція приймає два аргументи: `horizon` (горизонт прогнозування) і `season_period` (період сезонності).

Спочатку функція викликає `auto_arima` із заданими параметрами, зокрема з параметром сезонності, де 1 це її відсутність, для підбору оптимальної моделі ARIMA на навчальній вибірці `ts_train`. Потім модель навчається на цій вибірці методом `fit`. Далі функція застосовує навчену модель для прогнозування на горизонт значень, на ту саму довжину, що й у тестовій вибірці, використовуючи метод `predict`. Результат зберігається у змінну `forecastarm1`.

Значення функції, що повертаються, - це середнє абсолютне відсоткове відхилення MAPE між прогнозами і тестовою вибіркою `ts_test`, а також масив прогнозу `forecastarm1`. Ці прогнози на виході функцій моделей, як вже було сказано, знадобляться в побудові гібридної моделі надалі.

Цей код реалізує функцію `ExpSmooth`, яка виконує експоненціальне згладжування часового ряду і будує прогноз за допомогою моделі `ExponentialSmoothing` з бібліотеки `statsmodels`.

```

def ExpSmooth():
    global sestype
    if sestype == None:

```

```

modeladd = exptype(ts_train, 'additive')
modelmul = exptype(ts_train, 'multiplicative')
if modeladd[1] < modelmul[1]:
    sestype = 'additive'
else:
    sestype = 'multiplicative'

if season_period == 1:
    model = ExponentialSmoothing(ts, seasonal_periods=None, trend='additive',
seasonal=sestype,
                                damped_trend=True).fit()
else:
    model = ExponentialSmoothing(ts, seasonal_periods=season_period,
trend='additive', seasonal=sestype,
                                damped_trend=True).fit()

forecast = model.predict(0, len(ts) + horizon - 1)
forexphyb = forecast
print(forecast.iloc[len(ts):, ])
forecast.iloc[len(ts):, ].to_csv("ExponentialSmoothing.csv")
forecast = forecast.reset_index(drop=True)
fig = plt.figure(figsize=(18, 8))
ax = fig.add_subplot()
ax.xaxis.set_major_locator(plt.MaxNLocator())
plt.plot(ts, label="Time Series")
plt.plot(forecast.iloc[len(ts):, ], label="Prediction")
plt.legend(loc="best")
plt.savefig("ExponentialSmoothing.png", bbox_inches='tight')
plt.show()
return forexphyb

```

Перша дія функції - перевірити значення змінної `sestype`. Якщо вона дорівнює `None`, тобто не виконувалися функції розглянуті раніше як `exptest` і `exptype`, і тип сезонності не було обрано, то функція обирає оптимальний тип сезонності між "additive" і "multiplicative" на основі результатів моделей у функції `exptype`. Якщо модель з "additive" сезонністю має менше значення інформаційного критерію Акаїке, то `sestype` встановлюють у "additive", інакше - у "multiplicative".

Потім функція створює модель `ExponentialSmoothing` із заданими параметрами, включно з обраним типом сезонності (`sestype`). Якщо `season_period` дорівнює 1, то сезонність не враховується, інакше вказується значення `season_period` як кількість періодів сезонності. Модель навчається на всьому часовому ряді `ts`.

Після навчання моделі функція застосовує її для прогнозування значень часового ряду на всю довжину ряду, плюс на `horizon` кроків уперед. Результат зберігається у змінну `forecast`. Прогнозні значення, починаючи з моменту закінчення вихідного часового ряду `ts`, зберігаються у змінній `forexphub`, які знадобляться у вираховуванні гібридного прогнозу.

Далі функція виводить прогнозні значення (`forecast.iloc[len(ts):, ]`), ті значення, які користувач і хотів отримати, та зберігає їх у файл "ExponentialSmoothing.csv", з яким користувач зможе, якщо захоче, ознайомитися в робочій папці, зазначеній на початку програми. Потім будується графік, на якому відображається вихідний часовий ряд (`ts`) і прогнозні значення. Графік зберігається у файл "ExponentialSmoothing.png" і відображається на екрані.

Нарешті, функція повертає змінну `forexphub`, що містить прогнозні значення для наступного періоду.

```
def Arima():
```

```
    model = pm.auto_arima(ts, start_p=2, d=None, start_q=2, max_p=5,
```

```

max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2, max_D=1,
max_Q=2, max_order=5, m=season_period)

model.fit(ts)

forecast = model.predict(n_periods=horizon)

forarmhyb = forecast

print(forecast)

forecast.to_csv("ARIMA.csv")

forecast = forecast.reset_index(drop=True).to_frame()

empt = pd.DataFrame(index=range(len(ts)), columns=range(1))

forecast = pd.concat([empt, forecast], ignore_index=True)

fig = plt.figure(figsize=(18, 8))

ax = fig.add_subplot()

ax.xaxis.set_major_locator(plt.MaxNLocator())

plt.plot(ts, label="Time Series")

plt.plot(forecast, label="Prediction")

plt.legend(loc="best")

plt.savefig("ARIMA.png", bbox_inches='tight')

plt.show()

return forarmhyb

```

Цей код представляє функцію Arima, яка виконує аналіз часових рядів з використанням моделі ARIMA і генерує прогнозні значення.

Спочатку функція використовує функцію `auto_arima` з бібліотеки `rmдаріма` для автоматичного вибору оптимальних параметрів моделі ARIMA. Параметри `start_p`, `d`, `start_q`, `max_p`, `max_d`, `max_q`, `start_P`, `D`, `start_Q`, `max_P`, `max_D`, `max_Q`, `max_order` вказують діапазони значень параметрів моделі, які можуть бути протестовані для вибору оптимальних значень. Параметр `m` встановлюється рівним `season_period` для врахування сезонності, якщо вона

присутня. Модель підганяється до вихідного часового ряду `ts` за допомогою методу `fit`.

Потім модель використовується для генерації прогнозних значень на `horizon` кроків вперед за допомогою методу `predict`. Результат зберігається у змінній `forecast`. І знову ж таки, зберігаються в окрему змінну `forarmhyb`, для передачі прогнозних значень, які потрібні під час прорахунку гібридних значень надалі, якщо, звісно, буде завдання вирахувати додатково цю саму гібридну модель.

Далі функція виводить прогнозні значення `forecast` і зберігає їх у файл "`ARIMA.csv`", за аналогією як і в попередньому методі. Прогнозні значення також перетворюються у формат `DataFrame` і скидаються індекси. Створюється порожній `DataFrame` `empt` з розмірністю вихідного часового ряду `ts`, і потім прогнозні значення додаються у відповідний стовпець. Результат зберігається у змінну `forecast`.

Будується графік, на якому відображається вихідний часовий ряд `ts` і прогнозні значення `forecast`. Графік зберігається у файл "`ARIMA.png`" і відображається на екрані.

Наприкінці функція повертає змінну `forarmhyb`, що містить прогнозні значення.

Цей код представляє функцію `Hybrid`, яка виконує аналіз часових рядів з використанням гібридного підходу, комбінуючи прогнози, отримані за допомогою моделей `Exponential Smoothing` і `ARIMA`.

```
def Hybrid():
    global sestype
    if sestype == None:
        modeladd = exptype(ts_train, 'additive')
        modelmul = exptype(ts_train, 'multiplicative')
        if modeladd[1] < modelmul[1]:
```

```

    sestype = 'additive'
else:
    sestype = 'multiplicative'

if season_period == 1:
    modelexp = ExponentialSmoothing(ts, seasonal_periods=None, trend='additive',
seasonal=sestype,
                                damped_trend=True).fit()
else:
    modelexp = ExponentialSmoothing(ts, seasonal_periods=season_period,
trend='additive', seasonal=sestype,
                                damped_trend=True).fit()
modelarm = pm.auto_arima(ts, start_p=2, d=None, start_q=2, max_p=5,
                        max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2,
                        max_D=1, max_Q=2, max_order=5, m=season_period)
forecastexp = modelexp.predict(0, len(ts) + horizon - 1)
forecastarm = modelarm.predict(n_periods=horizon)
forecast1 = (forecastexp + forecastarm) / 2
forecastexp = forecastexp.reset_index(drop=True)
forecastexp = forecastexp.to_frame()
modelarm.fit(ts)
forecastarm = forecastarm.reset_index(drop=True).to_frame()
empt = pd.DataFrame(index=range(len(ts)), columns=range(1))
forecastarm = pd.concat([empt, forecastarm], ignore_index=True)
forecast = (forecastexp + forecastarm) / 2
print(forecast1.iloc[len(ts):, ])
forecast1.iloc[len(ts):, ].to_csv("Hybrid.csv")
fig = plt.figure(figsize=(18, 8))
ax = fig.add_subplot()
ax.xaxis.set_major_locator(plt.MaxNLocator())

```

```
plt.plot(ts, label="Time Series")
plt.plot(forecast.iloc[len(ts):, ], label="Prediction")
plt.legend(loc="best")
plt.savefig("Hybrid.png", bbox_inches='tight')
plt.show()
```

На початку функції перевіряється значення змінної `sestype`, якщо вже вона має тип сезонності, то такий тип і буде використовуватися надалі. Якщо воно дорівнює `None`, то проводиться вибір оптимального типу сезонності для моделі Exponential Smoothing. Для цього використовуються функції `exptype`, які підбирають моделі експоненціального згладжування з різними типами сезонності `additive` і `multiplicative` і повертають найкращу модель і значення інформаційного критерію AIC. Потім вибирається тип сезонності з найменшим значенням AIC і присвоюється змінній `sestype`.

Далі функція будує модель Exponential Smoothing і модель ARIMA. Якщо `season_period` дорівнює 1, то модель Exponential Smoothing будується без урахування сезонності, інакше встановлюється відповідне значення `seasonal_periods`. Обидві моделі підганяються до часового ряду `ts` за допомогою методів `fit`.

Далі відбувається генерація прогнозних значень для експоненціального згладжування `forecastexp` і ARIMA `forecastarm`. Прогнозні значення зберігаються у змінних `forecastexp` і `forecastarm` відповідно.

Далі виконується усереднення прогнозних значень. Створюється змінна `forecast1`, в яку записується усереднене значення між `forecastexp` і `forecastarm`. Потім у прогнозних значень скидаються індекси і йде перетворення у формат `DataFrame`. Перетворення початкової зміни прогнозу, зокрема перетворення в `DataFrame`, необхідне для того, щоб коректно виводився прогноз у графіку. Для прогнозних значень `arima` створюється ще додатково порожній `DataFrame` `empt` з розмірністю вихідного часового ряду `ts`, і прогнозні значення додаються у відповідний стовпець. Потім проводиться комбінування прогнозних значень

уже для нового формату прогнозу, результат якого зберігається у змінній `forecast`.

Функція виводить прогнозні значення `forecast1` і зберігає їх у файл `"Hybrid.csv"`. Потім будується графік, на якому відображається вихідний часовий ряд `ts` і прогнозні значення `forecast`. Графік зберігається у файл `"Hybrid.png"` і відображається на екрані.

Цей код містить блок умовних операторів, який ґрунтується на виборі користувача для аналізу часових рядів. Залежно від вибору, викликаються відповідні функції для виконання конкретного методу аналізу.

```
if choice == 1:  
    ExpSmooth()  
elif choice == 2:  
    Arima()  
elif choice == 3:  
    Hybrid()
```

Якщо значення змінної `choice` дорівнює 1, виконується блок коду `ExpSmooth()`. Це означає, що користувач обрав метод `Exponential Smoothing` для аналізу часового ряду. Викликається функція `ExpSmooth()`, яка містить код для побудови моделі `Exponential Smoothing` і генерації прогнозних значень.

Якщо значення змінної `choice` дорівнює 2, виконується блок коду `Arima()`. Це означає, що користувач вибрав метод `ARIMA` для аналізу часового ряду. Викликається функція `Arima()`, яка містить код для побудови моделі `ARIMA` і генерації прогнозних значень.

Якщо значення змінної `choice` дорівнює 3, виконується блок коду `Hybrid()`. Це означає, що користувач обрав гібридний метод для аналізу часового ряду. Викликається функція `Hybrid()`, яка містить код для комбінування прогнозних значень, отриманих за допомогою моделей `Exponential Smoothing` і `ARIMA`.

Кожен із блоків коду виконує відповідні операції, включно з побудовою моделей, генерацією прогнозних значень, збереженням результатів і побудовою графіків. Результати аналізу та графіки відображаються на екрані або зберігаються у відповідних файлах.

Цей код містить блок умовного оператора `elif` і виконує аналіз кількох моделей часових рядів для вибору найбільш підходящої моделі на основі значення MAPE (Mean Absolute Percentage Error).

```
elif choice == 4:
    expctest = expctest(horizon)
    print("ExpSmooth MAPE - ", expctest[0])
    sestype = expctest[2]
    armctest = armctest(horizon, season_period)
    print("ARIMA MAPE - ", armctest[0])
    forecasthyb = (expctest[1] + armctest[1]) / 2
    MAPEhyb = np.mean(np.abs((forecasthyb - ts_test.to_numpy() /
ts_test.to_numpy())) * 100
    print("Hybrid MAPE - ", MAPEhyb)

    set = [expctest[0], armctest[0], MAPEhyb]
    index = set.index(min(set))

    if index == 0:
        print("ExpSmooth має найменший MAPE, тому в якості основної моделі
було обрано цю модель.")
        ExpSmooth()
    elif index == 1:
        print("ARIMA має найменший MAPE, тому в якості основної моделі було
обрано цю модель.")
```

```
Arima()
```

```
else:
```

```
    print("Hybrid має найменший MAPE, тому в якості основної моделі було  
обрано цю модель.")
```

```
Hybrid()
```

Якщо значення змінної `choice` дорівнює 4, виконується блок коду, який виконує аналіз моделей Exponential Smoothing і ARIMA, а також гібридної моделі.

- Спочатку викликається функція `exptest(horizon)`, яка повертає результати аналізу моделі Exponential Smoothing. Виводиться значення MAPE цієї моделі, яку можна дістати з `exptest[0]`, а тип сезонності зберігається у змінну `sestype`.
- Потім викликається функція `armtest(horizon, season_period)`, яка виконує аналіз моделі ARIMA. Значення MAPE також виводиться, яке можна вже дістати вже звернувшись вже відповідно `armtest[0]`.
- Обчислюється гібридний прогноз шляхом усереднення прогнозів від моделей Exponential Smoothing і ARIMA. Тут використовуються вже ті значення прогнозів, які зберігали в окремі змінні і повертали її в кожній функції. Прогнозні значення зберігаються у змінній `forecasthyb`.
- Обчислюється MAPE для гібридного прогнозу і зберігається у змінну `MAPEhyb`. Виводяться значення MAPE гібридної моделі на екран.
- Створюється список `set`, що містить значення MAPE для кожної моделі.
- Визначається індекс моделі з найменшим значенням MAPE у списку `set` за допомогою `index = set.index(min(set))`.
- Залежно від значення індексу, виводиться повідомлення про обрану модель з найменшим MAPE і викликається відповідна функція для

виконання аналізу цієї моделі. Виклик кожної моделі було розглянуто на попередніх кроках відповідно.

У результаті цього блоку коду відбувається автоматичний вибір найбільш підходящої моделі на основі MAPE і виконання відповідного аналізу для обраної моделі.

У цьому коді використовується блок умовного оператора `elif`, де значення змінної `choice` дорівнює 5. Цей блок коду виконує аналіз трьох моделей часових рядів (Exponential Smoothing, ARIMA) і створює гібридний прогноз, усереднюючи прогнози від двох моделей.

```
elif choice == 5:
```

```
    forecastexp = ExpSmooth()
    forecastarm = Arima()
    forecast1 = (forecastexp + forecastarm) / 2
    forecastexp = forecastexp.reset_index(drop=True)
    forecastexp = forecastexp.to_frame()
    forecastarm = forecastarm.reset_index(drop=True).to_frame()
    empt = pd.DataFrame(index=range(len(ts)), columns=range(1))
    forecastarm = pd.concat([empt, forecastarm], ignore_index=True)
    forecast = (forecastexp + forecastarm) / 2
    print(forecast1.iloc[len(ts):, ])
    forecast1.iloc[len(ts):, ].to_csv("Hybrid.csv")
    fig = plt.figure(figsize=(18, 8))
    ax = fig.add_subplot()
    ax.xaxis.set_major_locator(plt.MaxNLocator())
    plt.plot(ts, label="Time Series")
    plt.plot(forecast.iloc[len(ts):, ], label="Prediction")
    plt.legend(loc="best")
    plt.savefig("Hybrid.png", bbox_inches='tight')
```

```
plt.show()
```

Якщо значення змінної `choice` дорівнює 5, виконується блок коду, який виконує такі дії:

Викликається функція `ExpSmooth()`, яка повертає прогностні значення для моделі Exponential Smoothing. Прогноз зберігається у змінну `forecastexp`.

Викликається функція `Arima()`, яка повертає прогностні значення для моделі ARIMA. Прогноз зберігається у змінну `forecastarm`.

Обчислюється гібридний прогноз шляхом усереднення прогнозів від моделей Exponential Smoothing і ARIMA. Прогностні значення зберігаються у змінній `forecast1`.

Перетворюються дані прогнозів у формат `DataFrame`. Прогноз моделі Exponential Smoothing зберігається у змінній `forecastexp`, а прогноз моделі ARIMA зберігається у змінній `forecastarm`.

Створюється порожній `DataFrame` `empt` з розмірністю, що відповідає вихідному часовому ряду. `DataFrame` `forecastarm` об'єднується з `empt` за допомогою методу `pd.concat()` для створення `DataFrame` із заповненими значеннями прогнозу.

Обчислюється гібридний прогноз, який буде виводитися графічним способом, шляхом усереднення прогнозів моделей Exponential Smoothing і ARIMA. Прогностні значення зберігаються у змінній `forecast`.

Виводяться значення прогнозу моделі Hybrid починаючи з моменту, а також цей прогноз моделі зберігається у файл "Hybrid.csv", за аналогією з попередніми збереженнями.

Створюється графік, на якому відображаються вихідний часовий ряд і прогноз моделі Hybrid. Графік зберігається у файл "Hybrid.png" і відображається на екрані.

У результаті цього блоку коду виконується аналіз трьох моделей часових рядів і виводиться гібридний прогноз, а також зберігаються відповідні файли та графіки.

У цьому коді використовується функція `input()` для очікування введення даних від користувача.

```
input('Натисніть ENTER, щоб вийти')
```

Викликається функція `input()`, яка виводить на екран повідомлення "Натисніть ENTER, щоб вийти" і очікує введення даних від користувача. Програма призупиняється і чекає, поки користувач натисне клавішу ENTER. Після натискання клавіші ENTER програма завершує своє виконання і виходить.

Таким чином, цей код пропонує користувачеві натиснути клавішу ENTER для завершення програми.

### **3.3 Висновок по третьому розділу**

Побудована система коду являє собою набір функцій і операцій для аналізу та прогнозування часових рядів. Вона містить кілька методів аналізу, таких як експоненціальне згладжування (Exponential Smoothing), модель ARIMA (Autoregressive Integrated Moving Average) і гібридний підхід, який об'єднує результати обох методів.

Основні кроки системи включають:

- Введення вхідних даних: Користувачеві пропонується ввести робочу папку, файл із часовим рядом, значення параметрів для аналізу часового ряду, такі як горизонт прогнозування і період сезонності.
- Перевірка та обробка пропущених значень: Система перевіряє часовий ряд на наявність пропущених значень. Цей крок важливий, оскільки пропущені значення можуть негативно вплинути на точність аналізу і прогнозування. Система застосовує відповідний метод обробки пропусків. Якщо пропущені значення виявлено, то користувачеві пропонується

застосувати один із методів обробки пропусків. Це може бути видалення пропущених значень, заповнення їх середніми значеннями, інтерполяція або інші методи відновлення даних.

- Виявлення аномалій: Якщо вказано сезонність, то відбувається процес декомпозиції ряду, де за допомогою аналізу залишкових значень, що ґрунтується на правилі 3 сигм, виокремлюють аномалії та пропонується користувачеві нівелювати їх.
- Вибір методу аналізу: Користувачеві пропонується вибрати один із запропонованих методів аналізу: експоненціальне згладжування, модель ARIMA, гібридний підхід, автоматичний вибір кращого методу або ж провести аналіз одразу на трьох методах.
- Застосування обраного методу: Залежно від обраного користувачем методу, відповідна функція викликається для аналізу та прогнозування часового ряду. Наприклад, якщо обрано експоненціальне згладжування, викликається функція `ExpSmooth()`, якщо обрано модель ARIMA, викликається функція `Arima()`, а якщо обрано гібридний підхід, викликається функція `Hybrid()`. Автоматичний вибір проводить моделювання всіх трьох методів на навчальній вибірці часового ряду, де пізніше обирає найкращий метод на основі найменшого MAPE. І вже обрану модель використовує на всьому часовому ряду.
- Результати прогнозування: Отримані прогнози відображаються на графіку разом із вихідним часовим рядом із використанням бібліотеки для візуалізації даних, такої як `Matplotlib`, що дає змогу візуально порівняти прогнози з реальними даними. Також отриманий графік і числові значення прогнозу зберігаються у відповідні файли в робочій папці.
- Завершення програми: Користувачеві пропонується натиснути клавішу `ENTER` для завершення програми.

Таким чином, побудована система надає гнучкий та інтерактивний інструмент для аналізу та прогнозування часових рядів, даючи змогу вибрати

найбільш підходящий метод і отримати точні прогнози для подальшого ухвалення рішень.

## **РОЗДІЛ 4. ДЕМОНСТРАЦІЯ ФУНКЦІОНАЛУ СИСТЕМИ ПРОГНОЗУВАННЯ МЕТОДАМИ ЧАСОВИХ РЯДІВ**

### **4.1 Вступ до розділу**

У цьому розділі буде продемонстровано функціонал розробленої системи автоматизованого прогнозування часових рядів. Буде завантажено тестові дані та застосовано різні методи прогнозування, доступні в нашій програмі, щоб побудувати прогнози на основі цих даних.

Для тестування функціоналу системи буде використано набір тестових даних, які являють собою часовий ряд з відомими значеннями протягом певного періоду часу. Це дає нам змогу перевірити точність та ефективність роботи системи.

Розроблена система надає кілька методів прогнозування, включно з експоненціальним згладжуванням, autoArima. Користувач може вибрати будь-який з доступних методів, для отримання результату у вигляді прогнозу часового ряду.

Буде застосовано кожен із цих методів до завантажених тестових даних. Крім того, буде отримано візуалізацію отриманих прогнозів за допомогою графіків, щоб наочно представити результати роботи системи.

У результаті буде продемонстровано, що система здатна ефективно прогнозувати часові ряди з використанням різних методів. Такий функціонал дає змогу підприємствам і організаціям ухвалювати обґрунтовані рішення на основі надійних прогнозів, що є важливим інструментом для управління та планування в бізнесі.

### **4.2 Використання системи**

Перед тим, як розпочати роботу із системою автоматизованого прогнозування часових рядів, створимо папку, після чого оголосимо її робочою папкою в програмі. Важливо зазначити, що робочу папку можна оголосити вже

існуючою в системі, не створюючи нової, але створимо нову для зручності використання. Робоча папка слугуватиме центральним місцем для зберігання всіх необхідних файлів і вихідних даних, пов'язаних із нашою програмою.

Робоча папка (working directory) у програмуванні являє собою директорію, в якій поточний процес або програма виконує свою роботу. Вона слугує основним місцем, де відбувається читання, запис і обробка файлів, а також доступ до інших ресурсів, пов'язаних із виконанням програми. Усі необхідні бібліотеки, модулі та скрипти можуть бути розміщені в цій папці для зручності управління та оновлення.

У робочій папці створимо файл із часовим рядом, який буде використовуватися в програмі. Цей файл містить історичні дані часового ряду, на основі яких буде будуватися прогноз. Важливо зберігати вихідні дані в робочій папці, щоб вони були легко доступні для обробки системою.

Крім того, робоча папка буде використовуватися для збереження вихідних даних із системи. Це включає в себе прогнозовані значення і графіки, отримані в результаті роботи системи. Зберігання вихідних даних у робочій папці забезпечує їхню структурованість і зручність доступу.

Таким чином, робоча папка є важливою складовою системи автоматизованого прогнозування часових рядів. Вона забезпечує зручне зберігання і доступ до вихідних даних, а також структуроване збереження результатів роботи системи, що полегшує аналіз і прийняття рішень на основі прогнозів.

Представимо роботу програми на часовому ряді, який показує кількість авіапасажирів на місяць, за певний період часу [43]. Цей датасет є дуже популярним, його часто розглядають у навчальних цілях із вивчення часових рядів. Оскільки такий датасет є вже класикою, то це чудовий варіант на ньому показати роботу системи.

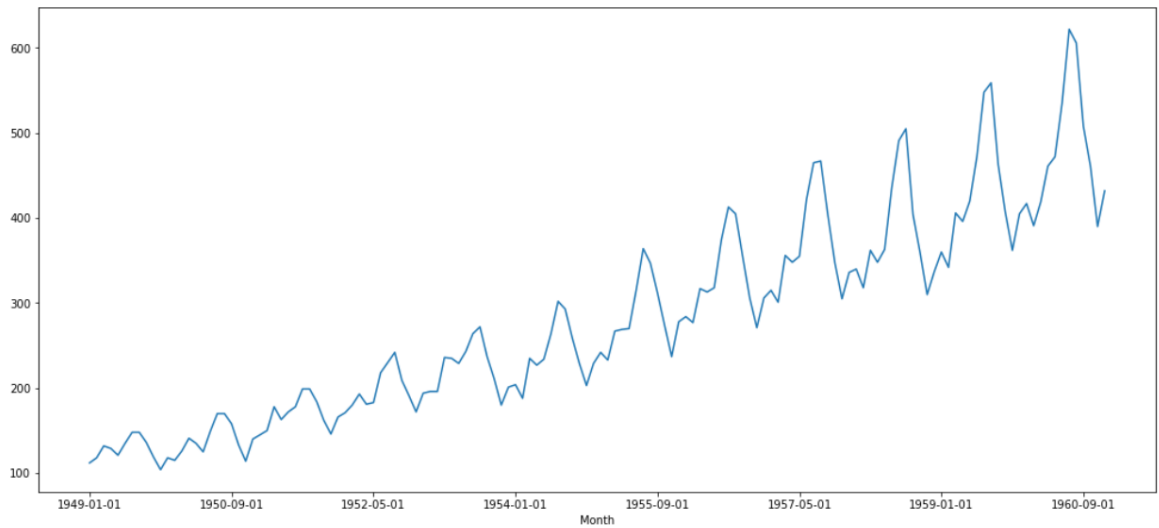


Рисунок 3.1 - Візуалізація вхідного часового ряду

Створимо нову папку.

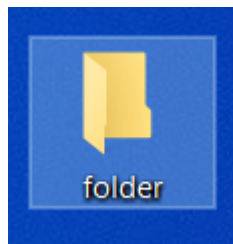


Рисунок 3.2 - Створили нову папку

Перенесемо в неї файл із вхідним часовим рядом.

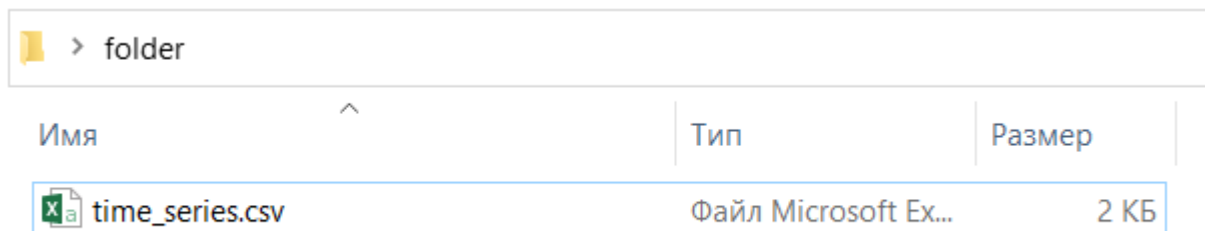


Рисунок 3.3 - Файл із часовим рядом у папці

Запускаємо програму.

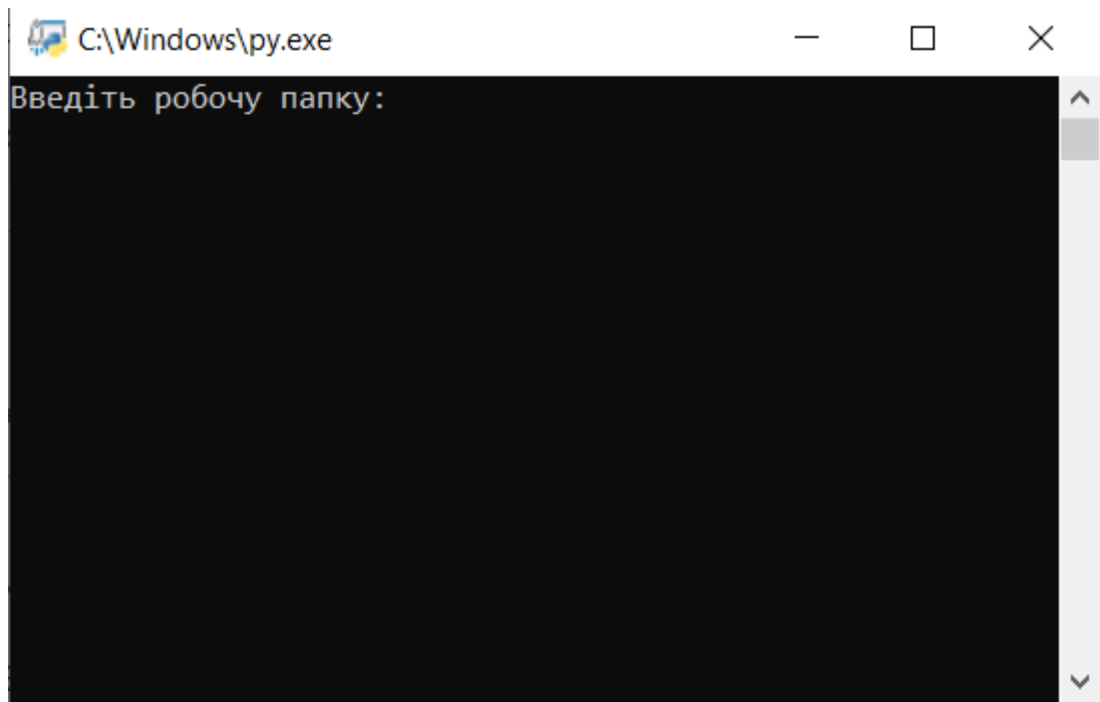


Рисунок 3.4 - Запущена програма в консолі операційної системи

Далі вивід роботи програми буде здійснюватися з консолі PyCharm для зручності.

Вводимо шлях до нашої створеної папки, яку хочемо оголосити робочою.

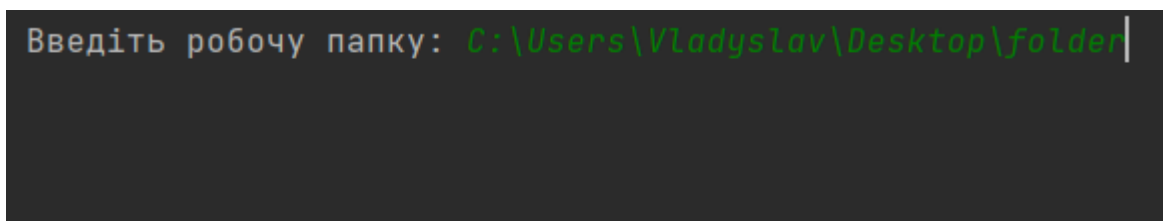
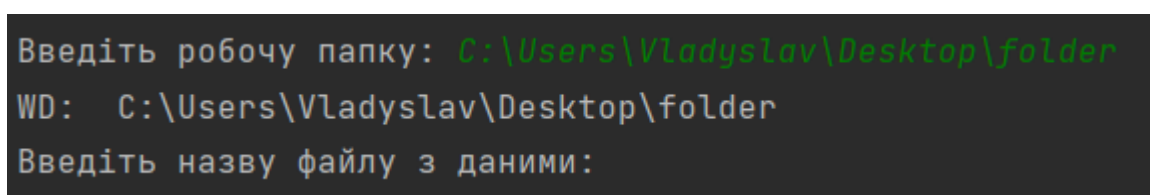


Рисунок 3.5 - Введення робочої папки

Програма виводить що робоча папка оголошена, і вимагає ввести назву файлу з даними.



### Рисунок 3.6 - Робочу папку оголошено

Вводимо повну назву файлу з часовим рядом.

```
WD: C:\Users\Vladyslav\Desktop\folder
Введіть назву файлу з даними: time_series.csv
На скільки значень вперед ви хочете побудувати прогноз? |
```

### Рисунок 3.7 - Дані завантажено

Жодних пропущених значень програма в ряду не виявила, як могли бачити раніше графік ряду, то так і є. І запитує далі про горизонт прогнозу.

Для прикладу роботи системи щодо заповнення пропущених значень, приберемо деякі значення з початкового ряду, створимо пропуски в ряду.

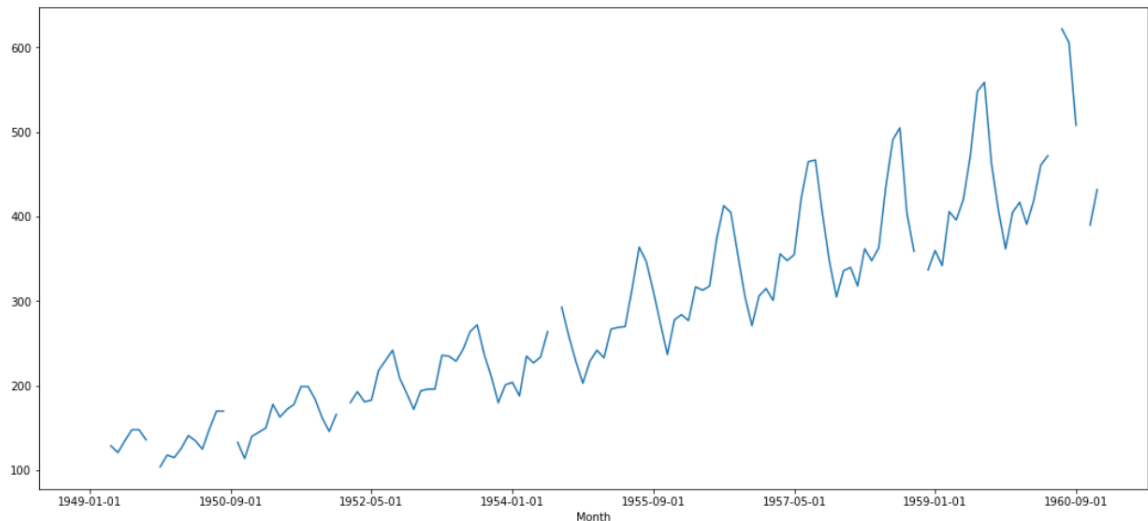


Рисунок 3.8 - Змінений часовий ряд, з пропущеними значеннями

Подивимося, як система цього разу поведеться.

```
Введіть робочу папку: C:\Users\Vladyslav\Desktop\folder
WD: C:\Users\Vladyslav\Desktop\folder
Введіть назву файлу з даними: time_series_with_missing_values.csv
У часовому ряду виявлено пропущені значення. Яким способом ви хочете їх виправити?
Доступні такі варіанти як: mean, median, locf, nobs, linear, spline, delete.
Напишіть, який варіант хочете використати: |
```

Рисунок 3.9 - У часовому ряду виявлено пропущені значення

Як можемо бачити, програма виявила пропущені значення. Програма запитує в який спосіб користувач хоче виправити їх, і надає варіанти, якими користувач може скористатися для вирішення цього завдання. Усі варіанти розв'язання були розглянуті раніше в попередньому розділі.

Скористаємося лінійною інтерполяцією для заповнення пропущених значень. Для цього введемо "linear" і натискаємо enter.

```
У часовому ряду виявлено пропущені значення. Яким способом ви хочете їх виправити?
Доступні такі варіанти як: mean, median, locf, nobs, linear, spline, delete.
Напишіть, який варіант хочете використати: linear
На скільки значень вперед ви хочете побудувати прогноз? |
```

Рисунок 3.10 - Застосування лінійної інтерполяції для заповнення пропущених значень

Як можемо бачити, програма застосувала варіант, який ввів користувач, і запитує про значення прогнозу часового ряду.

Функціонал із виявлення та заповнення пропущених значень у часовому ряду представлено. Повернемося до початкового, незміненого часового ряду.

Після того як цей часовий ряд було завантажено, і пройшла перевірка на пропущені значення, система запитує, на скільки значень вперед хочемо зробити прогноз.

Введемо для прикладу значення 12, що означає, що хочемо зробити прогноз на рік вперед, оскільки в часовому ряду значення йдуть щомісяця.

```
На скільки значень вперед ви хочете побудувати прогноз? 12
Яка кількість сезонних періодів у часовому ряду? (введіть значення 1 якщо сезонності немає) |
```

Рисунок 3.11 - Введено значення для горизонту прогнозу

Далі система запитує який є сезонний період, ввести 1 якщо сезонності немає. Як вже було сказано, у нас значення ряду йдуть щомісяця, тому вводимо значення 12.

```
Яка кількість сезонних періодів у часовому ряду? (введіть значення 1 якщо сезонності немає) 12
#Passengers
Month
1959-03-01      406
Виявлено можливі викиди в часовому ряду. Замінити їх на більш відповідні значення? (Так/Ні) |
```

Рисунок 3.12 - Значення сезонного періоду введено

Далі система проводить декомпозицію часового ряду, якщо вказана сезонність. І після виявляються можливі аномалії в часовому ряду. Як можемо бачити, система вивела можливий викид, його значення та повідомляє користувача про те, що можливі аномалії виявлено. Далі система запитує, чи потрібно ці можливі аномалії замінити на більш відповідні значення.

Програма виявила можливі викиди, за заданим алгоритмом, який їй задано, як було сказано раніше, тут використовувалося правило трьох сигм. Сама функція детекції аномалій вже являє собою важливий функціонал, який дає змогу аналітику визначити можливі аномалії в часовому ряду. А варто їх замінювати чи ні, то це вже може вирішити сам аналітик, знаючи більший контекст базового часового ряду.

Застосуємо тут функцію із заміни аномалій на більш відповідні значення. Введемо тут "Так".

```
Виявлено можливі викиди в часовому ряду. Замінити їх на більш відповідні значення? (Так/Ні) Так
Який метод аналізу часових рядів ви хочете застосувати?
Доступні такі варіанти як:
1) ExponentialSmoothing;
2) ARIMA;
3) Hybrid (середнє значення двох попередніх методів);
4) Автоматичний вибір;
5) Вивести всі три методи.
Введіть номер варіанта, який хочете застосувати:
```

Рисунок 3.13 - Функцію із заміни аномалій на більш відповідні значення застосовано

Далі у користувача програма запитує, який метод побудови прогнозу часового ряду він хоче застосувати. Як вже було описано, представляється такий список варіантів дії:

1. ExponentialSmoothing;
2. ARIMA;
3. Hybrid (середнє значення двох попередніх методів);
4. Автоматичний вибір;
5. Вивести всі три методи.

Користувач може, на свій розсуд, ввести номер варіанта, який відповідає певному методу. Модель будуватиме прогноз по всьому часовому ряду. Також користувач програми може вивести всі три методи, тоді методи будуть представлені по черзі, по порядку як вони виведені в списку.

Виберемо варіант з автоматичним вибором. Він вибере, як уже було обговорено, найкращий метод для побудови прогнозу, на навчальній вибірці цього часового ряду за метрикою MAPE. Введемо відповідний номер 4.

```
Який метод аналізу часових рядів ви хочете застосувати?  
Доступні такі варіанти як:  
1) ExponentialSmoothing;  
2) ARIMA;  
3) Hybrid (середнє значення двох попередніх методів);  
4) Автоматичний вибір;  
5) Вивести всі три методи.  
Введіть номер варіанта, який хочете застосувати: 4
```

Рисунок 3.14 - Вибір автоматичного методу

Далі програма запустить кілька методів підряд, для обчислення значення MAPE. Це може зайняти деякий час, йде обчислення трьох моделей. Варто зазначити, що з досвіду використання, метод ARIMA займає більший час для побудови моделей, в той же час метод експоненціального згладжування набагато швидше виконує обчислення.

```
ExpSmooth MAPE - 18.011279955785877  
ARIMA MAPE - 15.973667477111869  
Hybrid MAPE - 16.88673288556948  
ARIMA має найменший MAPE, тому в якості основної моделі було обрано цю модель.
```

Рисунок 3.15 - Вибір моделі з найменшим значенням MAPE

Як можемо бачити, виводиться значення MAPE для трьох моделей. У цьому випадку, з цим часовим рядом, метод ARIMA має найменше значення MAPE. Виходячи з цього, ця модель і вибирається як основна. Тепер будується прогноз часового ряду методом, який був обраний як основний, але вже на всьому початковому ряду. Після того як основний метод побудує прогноз, він виведе візуалізацію часового ряду разом із прогнозними значеннями.

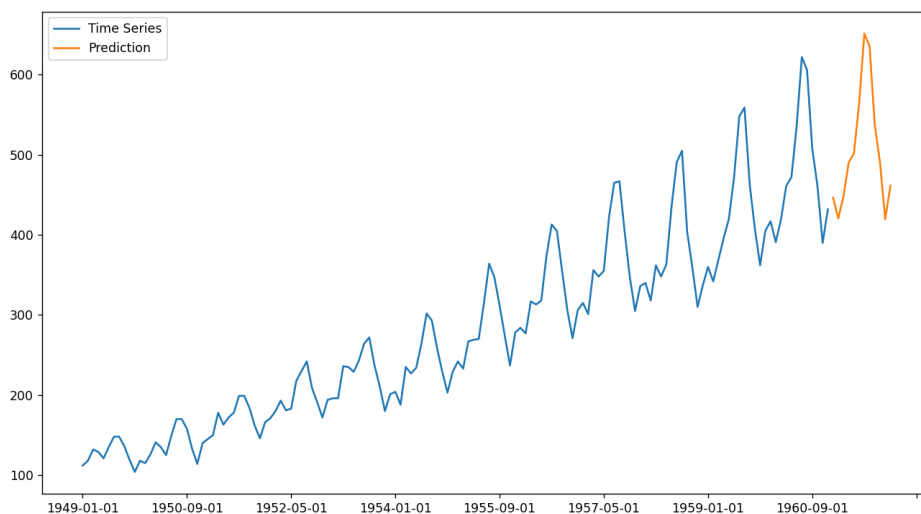


Рисунок 3.16 - Графік із прогностними значеннями та початковим часовим рядом

Щойно процедуру буде закінчено, користувачеві буде показано графік в окремому вікні, з прогностними значеннями і початковим часовим рядом. Як можемо бачити, програма побудувала прогноз за тими значеннями, які були введені раніше, і візуалізувала його разом з початковими даними.

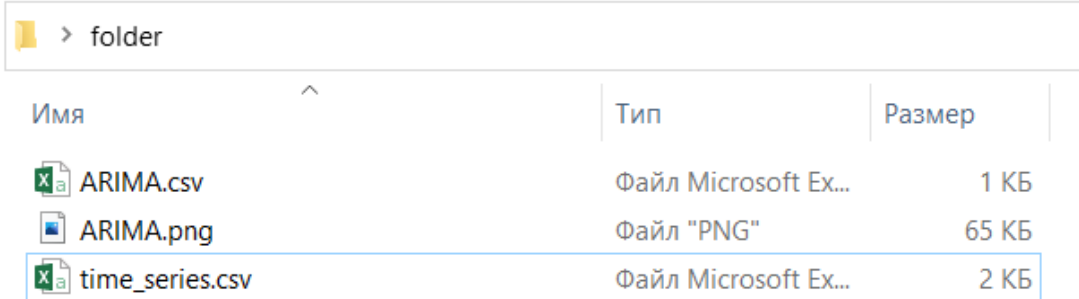
```

1961-01-01    446.541005
1961-02-01    420.541005
1961-03-01    448.541005
1961-04-01    490.541005
1961-05-01    501.541005
1961-06-01    564.541005
1961-07-01    651.541005
1961-08-01    635.541005
1961-09-01    537.541005
1961-10-01    490.541005
1961-11-01    419.541005
1961-12-01    461.541005
Freq: MS, dtype: float64
Натисніть ENTER, щоб вийти

```

Рисунок 3.17 - Числові значення прогнозу

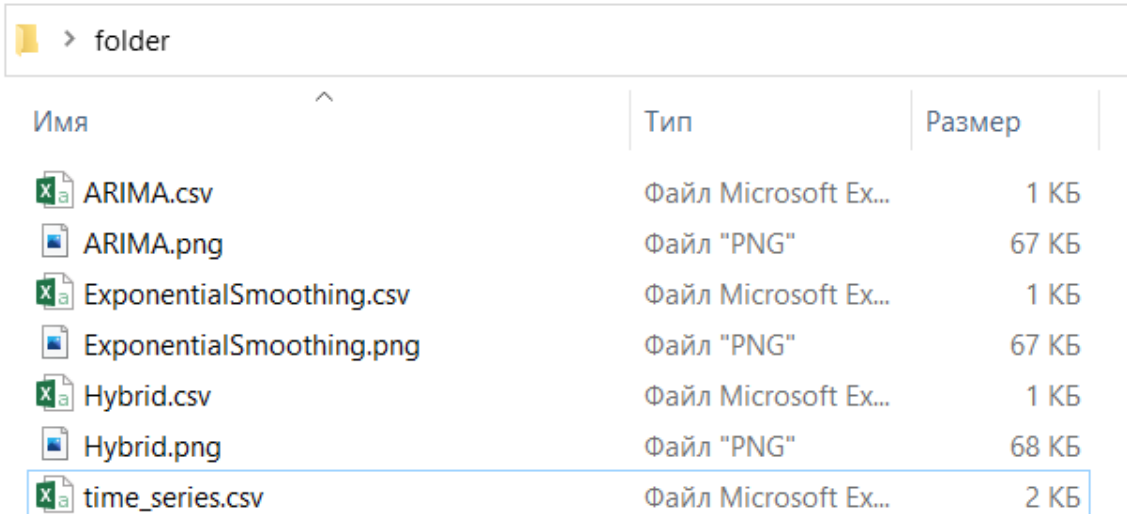
Також числові значення прогнозу виводяться в самому додатку, з якими користувач може ознайомитися окремо. Після закриття візуалізації з'являється рядок "Натисніть ENTER, щоб вийти", який дає змогу користувачеві завершити роботу програми.



Имя	Тип	Размер
ARIMA.csv	Файл Microsoft Ex...	1 КБ
ARIMA.png	Файл "PNG"	65 КБ
time_series.csv	Файл Microsoft Ex...	2 КБ

Рисунок 3.18 - Робоча папка з результатами прогнозу, після виконання програми

Також можемо повернутися до робочої папки, в якій знаходяться створені програмою файли, у вигляді png картинки графіка і csv файлу з числовими значеннями прогнозу, представлених раніше.



Имя	Тип	Размер
ARIMA.csv	Файл Microsoft Ex...	1 КБ
ARIMA.png	Файл "PNG"	67 КБ
ExponentialSmoothing.csv	Файл Microsoft Ex...	1 КБ
ExponentialSmoothing.png	Файл "PNG"	67 КБ
Hybrid.csv	Файл Microsoft Ex...	1 КБ
Hybrid.png	Файл "PNG"	68 КБ
time_series.csv	Файл Microsoft Ex...	2 КБ

Рисунок 3.19 - Робоча папка якщо було побудовано три моделі

Якщо користувач обере побудувати одразу всі три моделі, або ж він обере побудувати прогноз за кожною моделлю окремо. То результати прогнозних значень записуються в робочу папку окремо, кожна модель має свої файли. Варто зауважити, що файли перезаписуються, на вже нові значення, після нового запуску програми.

### **4.3 Висновок по четвертому розділу**

На закінчення цього розділу, було розглянуто функціонал системи автоматизованого прогнозування часових рядів. Було продемонстровано, як система дає змогу завантажувати дані, проводити попереднє оброблення, будувати прогнози з використанням різних методів прогнозування та зберігати результати.

Описаний функціонал системи є потужним інструментом для аналізу часових рядів і прийняття рішень на основі прогнозних даних. Він дає змогу автоматизувати процес прогнозування та значно скоротити час і зусилля, необхідні для аналізу даних і побудови прогнозів.

Система має низку переваг, включно з можливістю завантаження даних, проведення попередньої обробки, вибір і застосування різних методів прогнозування, візуалізацію прогнозу часових рядів, а також збереження результатів для подальшого аналізу.

Така система має широкий спектр застосування в різних сферах бізнесу та науки, де прогнозування часових рядів відіграє важливу роль. Вона може бути використана для прогнозування продажів, управління запасами, фінансового планування, аналізу ринку, операційного планування та багато чого іншого.

У подальшому розвитку системи можна розглянути можливість розширення функціоналу, наприклад, додавання додаткових методів прогнозування, а також приділення уваги поліпшенню користувацького інтерфейсу.

Загалом, функціонал системи надає користувачеві зручний та ефективний інструмент для роботи з часовими рядами, а його автоматизований підхід спрощує та прискорює процес аналізу та прогнозування.

## ВИСНОВКИ

У дипломній роботі було досліджено теоретичні засади аналізу часових рядів. Це включало вивчення основних понять, методів і підходів, що застосовуються для аналізу часових рядів. Було розглянуто такі концепції, як тренд, сезонність, випадковий шум, автокореляція і стаціонарність у часових рядах. Були розглянуті різні сімейства методів прогнозування часових рядів, від простого експоненціальне згладжування до метода ARIMA (авторегресійна інтегрована ковзаюча середня) із сезонною складовою.

Також було розглянуто приклади застосування аналізу часових рядів у бізнесі. Часові ряди можуть бути використані для прогнозування попиту на товари та послуги, управління запасами, оптимізації виробничих процесів і прийняття рішень у фінансовій сфері. Вони надають цінну інформацію про минулі тренди та дають змогу робити прогнози майбутніх подій, що допомагає бізнесу бути більш ефективним і конкурентоспроможним.

Була проведена літературна ревізія за темою часових рядів. Під час дослідження було розглянуто основні методи аналізу та прогнозування часових рядів, такі як ARIMA і експоненціальне згладжування. Було розглянуто їх застосування в різних сферах, включно з фінансовою аналітикою, економікою та охороною здоров'я. Кожна з цих моделей має свої плюси і мінуси, і що вони можуть по-різному змодельовати потрібний прогнозний результат. Також було розглянуто гібридні моделі, які при комбінуванні дають змогу врахувати в собі переваги різних моделей.

Важливим етапом дослідження було вивчення попередньої обробки часового ряду перед аналізом. Було розглянуто методи обробки пропущених даних, викидів. Попереднє опрацювання часових рядів відіграє ключову роль в отриманні точних прогнозів і поліпшенні якості моделей.

Іншим важливим аспектом, розглянутим у роботі, була декомпозиція часового ряду і виявлення аномалій. Декомпозиція дає змогу виокремити тренди, сезонні компоненти та залишки в часовому ряду, що сприяє глибшому

аналізу даних. Виявлення та обробка аномалій є важливим завданням в аналізі часових рядів, оскільки аномалії можуть спотворювати результати та призводити до неточних прогнозів. Застосування методу STL для виявлення аномалій та різних методів заповнення пропущених значень може значно покращити ту саму точність прогнозування.

Крім того, оцінка якості прогнозування є важливим етапом дослідження. Різні методи оцінки, такі як MAPE, можуть допомогти в оцінці якості моделі. Важливо зазначити, що іноді краще використовувати простішу модель, ніж складнішу, яка може не завжди давати кращий результат.

У роботі було також розглянуто проблему автоматизації в прогнозуванні часових рядів. Було досліджено методи автоматичного підбору параметрів та автоматизації процесу прогнозування. Автоматизація дає змогу проводити автоматизований аналіз, економить час і знижує ризик через суб'єктивний підхід аналітика. Однак ручний аналіз може показувати все ще більш конкурентоспроможний результат.

У результаті дослідження можна зробити висновок, що аналіз і прогнозування часових рядів є потужними інструментами для прийняття стратегічних рішень у бізнесі. Вибір конкретного методу прогнозування залежить від характеристик досліджуваних даних і поставлених цілей. Однак, незалежно від обраного методу, важливо враховувати особливості часових рядів і правильно інтерпретувати отримані результати.

Також було відзначена актуальність розроблення автоматизованої системи прогнозування часових рядів. В умовах швидко мінливого бізнес-оточення, автоматизація процесу прогнозування стає все більш важливою. Це дає змогу заощадити час і ресурси, а також поліпшити якість прогнозів.

Інформаційне забезпечення прийняття управлінських рішень методами аналізу часових рядів є важливим елементом планування та прийняття управлінських рішень. Одним із ключових завдань у цьому процесі є прогнозування майбутніх значень часових рядів, таких як продажі, виробництво, трафік на сайті та інші. У рамках створення такого забезпечення, і

було прийнято рішення про створення системи автоматизованого прогнозування часових рядів.

Для створення інформаційного забезпечення та системи автоматизованого програмування було вивчено та вирішено використати такі інформаційні та програмні технології як мова програмування Python з її бібліотеками для аналізу даних, інтерактивний інструмент Jupyter Notebook, а також інтегроване середовище розробки PyCharm.

Наступний крок полягав у моделюванні системи автоматизованого прогнозування часового ряду з використанням мови програмування Python та її відповідних бібліотек, а також середовища програмування PyCharm.

У рамках розробленої системи було створено функції, які раніше було розглянуто та вивчено. Введення вихідних даних, що дає змогу користувачам ввести часові ряди для аналізу та прогнозування.

Для забезпечення якості даних, система включає функцію перевірки та обробки пропущених значень. Ця функція виявляє та обробляє відсутні значення, що допомагає запобігти спотворенню результатів аналізу та прогнозування. Програма надає варіанти обробки від заповнення середнім значенням і до застосування методу лінійної інтерполяції.

Одним із важливих аспектів, реалізованих у системі, є функція виявлення аномалій. Вона дає змогу виявити та обробити аномальні значення в часовому ряді, які можуть спотворювати результати та призводити до неточних прогнозів. Це допомагає користувачам отримати більш надійні та точні результати прогнозування. Для цього, як було вище написано, застосовується метод STL декомпозиції, а також правило трьох сигм.

Система також надає можливість вибору методу аналізу часових рядів. Користувачі можуть вибрати один із раніше вивчених методів, таких як ARIMA або експоненціальне згладжування, залежно від своїх потреб і характеристик досліджуваних даних. Є можливість застосувати комбінацію цих двох методів у вигляді гібридної моделі. А також застосувати автоматичний вибір, який вибере

найкращу модель за навчальними даними на підставі метрики MAPE, а далі застосує вже цю найкращу модель для всього ряду.

Використовуючи обраний метод, система обробляє часові ряди і генерує результати прогнозу. Користувачі можуть отримати прогнозні значення на заданий період часу у вигляді графіка, або числових значень, і використовувати їх для прийняття бізнес-рішень або планування дій. Результати прогнозів зберігаються в робочу папку, яку користувач вкаже на початку програми.

Розроблена система надає зручний та ефективний інструмент для аналізу та прогнозування часових рядів. Вона об'єднує різні функції, які були розглянуті в роботі, і дає змогу користувачам проводити аналіз і прогнозування в зручному й автоматизованому режимі.

Було реалізовано систему автоматизованого прогнозування часового ряду у вигляді консольного додатка. Додаток надає користувачеві інтерфейс для введення та аналізу часових рядів, а також отримання прогнозних результатів.

Під час роботи було представлено демонстрацію функціоналу системи на тестових даних. Було продемонстровано різні етапи аналізу та прогнозування часових рядів, включно з введенням вихідних даних, обробкою пропущених значень, виявленням аномалій, вибором методу аналізу та отриманням результатів прогнозу у вигляді графіків або числових значень, а також збереженням результатів.

Підбиваючи підсумки, розроблена система надає зручний та ефективний інструмент для аналізу та прогнозування часових рядів. Консольний застосунок дає змогу користувачам легко взаємодіяти з системою, вводити дані та отримувати прогнози на основі обраного методу. Функціонал системи охоплює важливі етапи попередньої обробки даних, виявлення аномалій і вибору оптимального методу аналізу.

На закінчення, дана дипломна робота представила всебічний огляд і дослідження основ часових рядів, методів їхнього прогнозування, а також актуальність створення автоматизованої системи з прогнозування часових рядів. Розроблена система демонструє функціональність і практичну застосовність у

розв'язанні проблем бізнесу, де прогнозування часових рядів є важливим фактором успіху та прийняття обґрунтованих рішень.

## СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Box, George.E.P. and Jenkins, Gwilym M.. Time Series Analysis: Forecasting and Control. : Holden-Day, 1976.
2. Engle, Robert F. “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation.” *Econometrica* 50, no. 4 (1982): 987–1007. <https://doi.org/10.2307/1912773>.
3. Hamilton, James Douglas. Time Series Analysis. : Princeton University Press, 1994.
4. Hyndman RJ Athanasopoulos G. Forecasting : Principles and Practice. Heathmont Vic: OTexts; 2014.
5. Brockwell, Peter J. and Davis, Richard A.. Introduction to Time Series and Forecasting. : Springer International Publishing, 2016.
6. Mills TC. “The foundations of modern time series analysis. Palgrave Macmillan, Basingstoke. (2011) <https://doi.org/10.1057/9780230305021>
7. Understanding ARIMA Models for Machine Learning – URL: <https://www.capitalone.com/tech/machine-learning/understanding-arima-models/>
8. F. Wang, M. Li, Y. Mei and W. Li, "Time Series Data Mining: A Case Study With Big Data Analytics Approach," in *IEEE Access*, vol. 8, pp. 14322-14328, 2020, doi: 10.1109/ACCESS.2020.2966553.
9. Amunugama, A. W. V. M., Abeysundara, H. T. K., & Dissanayaka, U. N. B. (2022). ARIMA and FBMAP approach for forecasting daily stock price in Colombo Stock Exchange, Sri Lanka. *Ceylon Journal of Science*, 51(3), 259–267. DOI: <http://doi.org/10.4038/cjs.v51i3.8033>
- 10.S. I. Ansari and V. H Y, “System for Analysis and Prediction of Trends in Cryptocurrency Market”, *EAI Endorsed Trans IoT*, vol. 7, no. 27, pp. 1–5, Mar. 2022.
- 11.Exponential Smoothing – URL: <https://www.vercast.eu/post/exponential-smoothing>

12. George Moiseev, Forecasting oil tanker shipping market in crisis periods: Exponential smoothing model application, *The Asian Journal of Shipping and Logistics*, Volume 37, Issue 3, 2021, Pages 239-244, ISSN 2092-5212, <https://doi.org/10.1016/j.ajsl.2021.06.002>
13. Didem Guleryuz, Forecasting outbreak of COVID-19 in Turkey; Comparison of Box–Jenkins, Brown’s exponential smoothing and long short-term memory models, *Process Safety and Environmental Protection*, Volume 149, 2021, Pages 927-935, ISSN 0957-5820, <https://doi.org/10.1016/j.psep.2021.03.032>
14. Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 27(3), 1–22. <https://doi.org/10.18637/jss.v027.i03>
15. Appaia, L., & Palraj, S. (2023). On replacement of outliers and missing values in time series. *EQA - International Journal of Environmental Quality*, 53(1), 1–10. <https://doi.org/10.6092/issn.2281-4485/16184>
16. W.R. Terry, J.B. Lee, A. Kumar, Time series analysis in acid rain modeling: Evaluation of filling missing values by linear interpolation, *Atmospheric Environment* (1967), Volume 20, Issue 10, 1986, Pages 1941-1943, ISSN 0004-6981, [https://doi.org/10.1016/0004-6981\(86\)90335-5](https://doi.org/10.1016/0004-6981(86)90335-5)
17. Ma S, Liu Q, Zhang Y. A prediction method of fire frequency: Based on the optimization of SARIMA model. *PLoS One*. 2021 Aug 9;16(8):e0255857. doi: 10.1371/journal.pone.0255857. PMID: 34370785; PMCID: PMC8352067.
18. Yuyun LIANG, Yao YAO, Xiaoqin YAN, et al. Estimating the Spatial Variation of Electricity Consumption Anomalies and the Influencing Factors [J]. *Journal of Geodesy and Geoinformation Science*, 2022, 5(2):29-37. DOI:10.11947/j.JGGS.2022.0204
19. Anomaly Detection in Time Series – URL: <https://neptune.ai/blog/anomaly-detection-in-time-series>
20. Suhartono, S. (2005). A COMPARATIVE STUDY OF FORECASTING MODELS FOR TREND AND SEASONAL TIME SERIES DOES COMPLEX MODEL ALWAYS YIELD BETTER FORECAST THAN

- SIMPLE MODELS . Jurnal Teknik Industri, 7(1), pp. 22-30.  
<https://doi.org/10.9744/jti.7.1.pp.22-30>
21. Kuan M-M. 2022. Applying SARIMA, ETS, and hybrid models for prediction of tuberculosis incidence rate in Taiwan. PeerJ 10:e13117 DOI 10.7717/peerj.13117
22. Safi, Samir K. and Sanusi, Olajide Idris. 'A Hybrid of Artificial Neural Network, Exponential Smoothing, and ARIMA Models for COVID-19 Time Series Forecasting'. 1 Jan. 2021 : 25 – 35
23. Alsharif, A.; Sonia; Kumar, K.; Iwendi, C. Time Series Data Modeling Using Advanced Machine Learning and AutoML. Sustainability 2022, 14, 15292. <https://doi.org/10.3390/su142215292>
24. What is Auto-ARIMA? – URL:  
<https://medium.com/featurepreneur/what-is-auto-arima-b8025c6d732d>
25. E.Sayed, Hanaa & A.Gabbar, Hossam & Fouadc, Soheir & Ahmedc, Khalil. A Forecasting Decision Support System. (2008).
26. Components of Time Series – URL:  
<https://www.toppr.com/guides/business-mathematics-and-statistics/time-series-analysis/components-of-time-series/>
27. Yugesh Verma, A Beginner's Guide to Characteristics of a Time-Series – URL:  
<https://analyticsindiamag.com/a-beginners-guide-to-characteristics-of-a-time-series/>
28. KSV Muralidhar, Understanding Autocorrelation in Time Series Analysis – URL:  
<https://towardsdatascience.com/understanding-autocorrelation-in-time-series-analysis-322ad52f2199>
29. Stationarity and differencing – URL:  
<https://people.duke.edu/~rnau/411diff.htm>
30. Deineha V., Mochalova D., ANALYSIS OF FORECASTING METHODS FOR FINANCIAL MARKETS, Information Technology and Implementation (IT&Is-2021)

31. Ostertagova, Eva & Ostertag, Oskar. Forecasting Using Simple Exponential Smoothing Method. *Acta Electrotechnica et Informatica*. 12. 62–66. (2012). 10.2478/v10198-012-0034-2.
32. Yibin Ng, Forecasting Stock Prices using Exponential Smoothing – URL: <https://towardsdatascience.com/forecasting-stock-prices-using-exponential-smoothing-b37dfe54e8e9>
33. Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. – URL: <https://otexts.com/fpp2/AR.html>
34. Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. – URL: <https://otexts.com/fpp2/MA.html>
35. Rushad Faridi, *Time Series Analysis* – URL: [https://bookdown.org/rushad\\_16/TSA\\_Lectures\\_book/general-arma-models.html](https://bookdown.org/rushad_16/TSA_Lectures_book/general-arma-models.html)
36. Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. – URL: <https://otexts.com/fpp2/non-seasonal-arma.html>
37. Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. – URL: <https://otexts.com/fpp2/seasonal-arma.html>
38. Jim Frost, *Autocorrelation and Partial Autocorrelation in Time Series Data* – URL: <https://statisticsbyjim.com/time-series/autocorrelation-partial-autocorrelation/>
39. Enes Zvornicanin, *Choosing the best q and p from ACF and PACF plots in ARMA-type modeling* – URL: <https://www.baeldung.com/cs/acf-pacf-plots-arma-modeling>
40. Deineha V., Khlevna I., PREDICTING FRAUDULENT CRYPTOCURRENCY TRANSACTIONS USING LOGISTIC

REGRESSION, International Journal of Information and Communication  
Technologies

41. What Is 3 Sigma in Statistics? (And How To Find It) – URL:

<https://www.indeed.com/career-advice/career-development/3-sigma>

42. How to Perform an Anti-Join in Pandas – URL:

<https://www.statology.org/pandas-anti-join/>

43. Kaggle platform, Air Passengers – URL:

<https://www.kaggle.com/datasets/rakannimer/air-passengers>

## ДОДАТКИ

### Додаток А

#### Програмний код

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot
import pmdarima as pm
from datetime import datetime
from datetime import timedelta
from pandas.plotting import register_matplotlib_converters
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.statespace.sarimax import SARIMAX
register_matplotlib_converters()
from time import time
from matplotlib import pyplot as plt, dates as mdates
from statsmodels.tsa.seasonal import STL
from statsmodels.tsa.api import ExponentialSmoothing

working_directory = input("Введіть робочу папку: ")
os.chdir(working_directory)
print("WD: ", os.getcwd())

file_name = input("Введіть назву файлу з даними: ")
ts = pd.read_csv(file_name)
ts.iloc[:,0] = pd.to_datetime(ts.iloc[:,0]).dt.strftime('%Y-%m-%d')
ts.set_index(ts.iloc[:,0],inplace = True)
ts.drop(ts.columns[0], axis=1, inplace=True)

nul_data = pd.isnull(ts.iloc[:,0])
```

```

if not ts[nul_data].empty:
    print("У часовому ряду виявлено пропущені значення. Яким способом ви
хочете їх виправити?")
    print("Доступні такі варіанти як: mean, median, locf, nocb, linear, spline,
delete.")
    preparation = input("Напишіть, який варіант хочете використати: ")
    if preparation == 'delete':
        ts = ts.dropna(axis = 0, how = 'any')
    if preparation == 'mean':
        ts.iloc[:,0] = ts.assign(FillMean=ts.iloc[:,0].fillna(ts.iloc[:,0].mean())).iloc[:,1]
    if preparation == 'median':
        ts.iloc[:,0] = ts.assign(FillMean=ts.iloc[:,0].fillna(ts.iloc[:,0].median())).iloc[:,1]
    if preparation == 'locf':
        ts.iloc[:,0] = ts.iloc[:,0].fillna(method = 'ffill')
    if preparation == 'nocb':
        ts.iloc[:,0] = ts.iloc[:,0].fillna(method = 'bfill')
    if preparation == 'linear':
        ts.iloc[:,0] = ts.iloc[:,0].interpolate(method='linear')
    if preparation == 'spline':
        ts.iloc[:,0] = ts.iloc[:,0].interpolate(option='spline')

horizon = int(input("На скільки значень вперед ви хочете побудувати прогноз? "))
season_period = int(input("Яка кількість сезонних періодів у часовому ряду?
(введіть значення 1 якщо сезонності немає) "))

if season_period != 1:
    stl = STL(ts.iloc[:,0], period = season_period)
    result = stl.fit()
    resid = result.resid

```

```

resid_mean = resid.mean()
resid_std = resid.std()

upper = resid_mean + 3*resid_std
lower = resid_mean - 3*resid_std

anomalies = ts[(resid < lower) | (resid > upper)]
anomalies = anomalies.reset_index()
outer = anomalies.merge(ts[nul_data], how='outer', indicator = True)
anomalies = outer[(outer._merge=='left_only')].drop('_merge', axis=1)
anomalies.set_index(anomalies.iloc[:,0], inplace = True)
anomalies.drop(anomalies.columns[0], axis=1, inplace=True)
print(anomalies)

if not anomalies.empty:
    anomquest = input("Виявлено можливі викиди в часовому ряду. Замінити їх
на більш відповідні значення? (Так/Ні) ")
    if anomquest=="Так" or anomquest=="так":
        i = 0
        while i < len(anomalies.index):
            ts[ts.index == anomalies.index[i]] = np.nan
            i = i + 1
        ts.iloc[:,0] = ts.iloc[:,0].interpolate(method='linear')

print("Який метод аналізу часових рядів ви хочете застосувати?")
print("Доступні такі варіанти як:")
print("1) ExponentialSmoothing;")
print("2) ARIMA;")
print("3) Hybrid (середнє значення двох попередніх методів);")
print("4) Автоматичний вибір;")

```

```

print("5) Вивести всі три методи.")
choice = int(input("Введіть номер варіанта, який хочете застосувати: "))

sestype = None
train_length = len(ts) - horizon
ts_train = ts.iloc[0:train_length]
ts_test = ts.iloc[train_length:len(ts)]

def exptype(x, y):
    if season_period == 1:
        model = ExponentialSmoothing(x, seasonal_periods=None, trend='additive',
seasonal=y, damped_trend = True).fit()
    else:
        model = ExponentialSmoothing(x, seasonal_periods=season_period, trend='
additive', seasonal=y, damped_trend = True).fit()
    return model, model.aic

def exptest(horizon):
    modeladd = exptype(ts_train, 'additive')
    modelmul = exptype(ts_train, 'multiplicative')
    if modeladd[1] < modelmul[1]:
        model = modeladd[0]
        sestype = 'additive'
    else:
        model = modelmul[0]
        sestype = 'multiplicative'

fitted = model.predict(0, len(ts_train) + horizon - 1)
forecastexp = fitted.iloc[len(ts_train):len(fitted)]
forecastexp1 = forecastexp.to_numpy()

```

```

ts_test1 = ts_test.to_numpy()
return (np.mean(np.abs((forecastxp1 - ts_test1)/ts_test1))*100), forecastxp1,
sestype

```

```

def armtest(horizon, season_period):
    model = pm.auto_arima(ts_train, start_p=2, d=None, start_q=2, max_p=5,
                          max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2,
                          max_D=1, max_Q=2, max_order=5, m=season_period)
    model.fit(ts_train)
    forecastarm = model.predict(n_periods=horizon)
    forecastarm1 = forecastarm.to_numpy().flatten()
    ts_test1 = ts_test.to_numpy()
    return (np.mean(np.abs((forecastarm1 - ts_test1)/ts_test1))*100), forecastarm1

```

```

def ExpSmooth():
    global sestype
    if sestype == None:
        modeladd = exptype(ts_train, 'additive')
        modelmul = exptype(ts_train, 'multiplicative')
        if modeladd[1] < modelmul[1]:
            sestype = 'additive'
        else:
            sestype = 'multiplicative'

    if season_period == 1:
        model = ExponentialSmoothing(ts, seasonal_periods=None, trend='additive',
seasonal=sestype,
                                damped_trend=True).fit()
    else:

```

```
model = ExponentialSmoothing(ts, seasonal_periods=season_period,
trend='additive', seasonal=sestype,
                             damped_trend=True).fit()
```

```
forecast = model.predict(0, len(ts) + horizon - 1)
forexphyb = forecast
print(forecast.iloc[len(ts):, ])
forecast.iloc[len(ts):, ].to_csv("ExponentialSmoothing.csv")
forecast = forecast.reset_index(drop=True)
fig = plt.figure(figsize=(18, 8))
ax = fig.add_subplot()
ax.xaxis.set_major_locator(plt.MaxNLocator())
plt.plot(ts, label="Time Series")
plt.plot(forecast.iloc[len(ts):, ], label="Prediction")
plt.legend(loc="best")
plt.savefig("ExponentialSmoothing.png", bbox_inches='tight')
plt.show()
return forexphyb
```

```
def Arima():
```

```
    model = pm.auto_arima(ts, start_p=2, d=None, start_q=2, max_p=5,
                          max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2,
                          max_D=1, max_Q=2, max_order=5, m=season_period)
    model.fit(ts)
    forecast = model.predict(n_periods=horizon)
    forarmhyb = forecast
    print(forecast)
    forecast.to_csv("ARIMA.csv")
    forecast = forecast.reset_index(drop=True).to_frame()
```

```

empt = pd.DataFrame(index=range(len(ts)), columns=range(1))
forecast = pd.concat([empt, forecast], ignore_index=True)
fig = plt.figure(figsize=(18, 8))
ax = fig.add_subplot()
ax.xaxis.set_major_locator(plt.MaxNLocator())
plt.plot(ts, label="Time Series")
plt.plot(forecast, label="Prediction")
plt.legend(loc="best")
plt.savefig("ARIMA.png", bbox_inches='tight')
plt.show()
return forarmhyb

```

```
def Hybrid():
```

```
    global sestype
```

```
    if sestype == None:
```

```
        modeladd = exptype(ts_train, 'additive')
```

```
        modelmul = exptype(ts_train, 'multiplicative')
```

```
        if modeladd[1] < modelmul[1]:
```

```
            sestype = 'additive'
```

```
        else:
```

```
            sestype = 'multiplicative'
```

```
    if season_period == 1:
```

```
        modelexp = ExponentialSmoothing(ts, seasonal_periods=None, trend='additive',
seasonal=sestype,
```

```
                                damped_trend=True).fit()
```

```
    else:
```

```
        modelexp = ExponentialSmoothing(ts, seasonal_periods=season_period,
trend='additive', seasonal=sestype,
```

```
                                damped_trend=True).fit()
```

```

modelarm = pm.auto_arma(ts, start_p=2, d=None, start_q=2, max_p=5,
                        max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2,
                        max_D=1, max_Q=2, max_order=5, m=season_period)
forecastexp = modelexp.predict(0, len(ts) + horizon - 1)
forecastarm = modelarm.predict(n_periods=horizon)
forecast1 = (forecastexp + forecastarm) / 2
forecastexp = forecastexp.reset_index(drop=True)
forecastexp = forecastexp.to_frame()
modelarm.fit(ts)
forecastarm = forecastarm.reset_index(drop=True).to_frame()
empt = pd.DataFrame(index=range(len(ts)), columns=range(1))
forecastarm = pd.concat([empt, forecastarm], ignore_index=True)
forecast = (forecastexp + forecastarm) / 2
print(forecast1.iloc[len(ts):, ])
forecast1.iloc[len(ts):, ].to_csv("Hybrid.csv")
fig = plt.figure(figsize=(18, 8))
ax = fig.add_subplot()
ax.xaxis.set_major_locator(plt.MaxNLocator())
plt.plot(ts, label="Time Series")
plt.plot(forecast.iloc[len(ts):, ], label="Prediction")
plt.legend(loc="best")
plt.savefig("Hybrid.png", bbox_inches='tight')
plt.show()

if choice == 1:
    ExpSmooth()
elif choice == 2:
    Arima()
elif choice == 3:
    Hybrid()

```

```

elif choice == 4:
    exptest = exptest(horizon)
    print("ExpSmooth MAPE - ", exptest[0])
    sestype = exptest[2]
    armtest = armtest(horizon, season_period)
    print("ARIMA MAPE - ", armtest[0])
    forecasthyb = (exptest[1] + armtest[1]) / 2
    MAPEhyb = np.mean(np.abs((forecasthyb - ts_test.to_numpy() /
ts_test.to_numpy())) * 100
    print("Hybrid MAPE - ", MAPEhyb)

    set = [exptest[0], armtest[0], MAPEhyb]
    index = set.index(min(set))

    if index == 0:
        print("ExpSmooth має найменший MAPE, тому в якості основної моделі
було обрано цю модель.")
        ExpSmooth()
    elif index == 1:
        print("ARIMA має найменший MAPE, тому в якості основної моделі було
обрано цю модель.")
        Arima()
    else:
        print("Hybrid має найменший MAPE, тому в якості основної моделі було
обрано цю модель.")
        Hybrid()
elif choice == 5:
    forecastexp = ExpSmooth()
    forecastarm = Arima()
    forecast1 = (forecastexp + forecastarm) / 2

```

```
forecastexp = forecastexp.reset_index(drop=True)
forecastexp = forecastexp.to_frame()
forecastarm = forecastarm.reset_index(drop=True).to_frame()
empt = pd.DataFrame(index=range(len(ts)), columns=range(1))
forecastarm = pd.concat([empt, forecastarm], ignore_index=True)
forecast = (forecastexp + forecastarm) / 2
print(forecast1.iloc[len(ts):, ])
forecast1.iloc[len(ts):, ].to_csv("Hybrid.csv")
fig = plt.figure(figsize=(18, 8))
ax = fig.add_subplot()
ax.xaxis.set_major_locator(plt.MaxNLocator())
plt.plot(ts, label="Time Series")
plt.plot(forecast.iloc[len(ts):, ], label="Prediction")
plt.legend(loc="best")
plt.savefig("Hybrid.png", bbox_inches='tight')
plt.show()
```

```
input('Натисніть ENTER, щоб вийти')
```