МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
в.о. завідувача кафедри
_____ Рязанцев О.І.
«____»_____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Методи аналізу та прогнозування бізнес даних

Освітній рівень"Магістр"
Спеціальність 122 "Комп'ютерні науки"

Науковий керівник роботи:	_____ (підпис)	О.І.Рязанцев (ініціали, прізвище)
Консультант з охорони праці:	_____ (підпис)	Я.О.Критська (ініціали, прізвище)
Студент:	_____ (підпис)	В.О.Даниленко (ініціали, прізвище)
Група:		КН-19дм

Сєвєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітній рівень магістр
Напрям підготовки _____
(шифр і назва)
Спеціальність 122 "Комп'ютерні науки"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача кафедри _____
_____ В.С.Кардашук
«_____» _____20____р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Даниленко Владиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи аналізу та прогнозування бізнес даних

керівник проекту (роботи) Рязанцев Олександр Іванович, д.т.н., проф.
(прізвище, м.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «5» 10 2020 р. № 140/15.15
2. Строк подання студентом роботи 10.01.2021
3. Вихідні дані до роботи Матеріали науково-дослідної практики, теоретичні відомості про методи аналізу та прогнозовування часових рядів, програмна реалізація та порівняння результатів прогнозування

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області, моделі прогнозування часових рядів, програмна реалізація моделей прогнозування часових рядів, охорона праці та безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 14.10.2020

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Розробка технічного завдання	02.09.2020-15.09.2020	
2	Аналіз літератури з досліджуваної проблеми	16.09.2020-22.09.2020	
3	Аналіз технічних засобів та розробка методу	23.09.2020-25.09.2020	
4	Програмна реалізація	26.09.2020-06.10.2020	
5	Аналіз результатів дослідження	07.10.2020-25.11.2020	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2020-1.12.2020	
7	Оформлення пояснювальної записки, автореферату та презентації	2.12.2020-09.01.2021	

Студент _____ Даниленко В.О.
(підпис) (прізвище та ініціали)

Науковий керівник _____ Рязанцев О.І.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Даниленко В.О. Методи аналізу та прогнозування бізнес даних.

Метою магістерської роботи є дослідження та розробка методів прогнозування часових рядів.

Об’єктом дослідження є часові ряди та моделі для їх прогнозування.

У процесі виконання роботи було використано методи ARMA, ARIMA, SARIMA та Холта-Вінтерса для аналізу та прогнозування. Було досліджено основні складові часового ряду, були розглянуто методи для перевірки даних на стаціонарність та у разі нестаціонарних даних як перетворити їх на стаціонарні. Дані методи було протестовано на трьох різних наборах даних. Результати роботи кожного з методів були порівняні. Прогнози були досліджені на помилки за допомогою методів середнього квадрату помилки розподілу та середньоквадратичного відхилення.

Результатом роботи є комп’ютерна програма для роботи з CRM системою, яка дозволяє користувачеві підключатися до CRM системи для отримання даних які будуть спрогнозовані, або самостійно загрузити csv файл та виконати прогнозування.

Ключові слова: часові ряди, arma, arima, sarima, холт-вінтерс, експоненційне згладжування, авторегресія, ковзне середнє, критерій акаіке, mse, rmse.

ABSTRACT

Danilenko V.O. Methods of analysis and forecasting of business data.

The purpose of this work is to research and develop methods for predicting time series.

The object of study is time series and models for predicting time series.

ARMA, ARIMA, SARIMA and Holt-Winters methods for analysis and forecasting were used in the course of the work. The main components of the time series were investigated, methods for verifying stationary data and, in the case of non-stationary data, how to convert them to stationary were considered. These methods have been tested on three different datasets. The results of each method were compared. The forecasts were investigated for errors using the methods of the mean squared error and root mean squared error.

The result is a CRM application that allows the user to connect to the CRM system to obtain data that will be predicted, or to upload the csv file independently and perform forecasting.

Keywords: time series, arma, arima, sarima, holt-winters, exponential smoothing, autoregression, moving average, akaike criteria, mse, rmse

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

MSE – mean squared error (середньоквадратична помилка розподілу)

RMSE – root mean squared error (середньоквадратичне відхилення)

IoT – internet of things (інтернет речей)

MAPE – mean absolute percentage error (середня абсолютна процентна помилка)

MPE – mean percentage error (середній процент відхилення)

PE – percentage error (процентна помилка)

SSE – sum of squared errors (сума квадрату помилок)

MAE – mean absolute error (середня абсолютна помилка)

ME – mean error (середня помилка)

ACF – autocorrelation function (функція автокореляції)

PACF – partial autocorrelation function (функція часткової автокореляції)

DF – Dickey – Fuller test (тест Діккі – Фуллера)

ADF – augmented Dickey – Fuller test (розширений тест Діккі – Фуллера)

AIC – Akaike information criterion (інформаційний критерій Акаіке)

BIC – Bayesian information criterion (інформаційний критерій Баєса)

Q–Q – quantile–quantile plot (графік квантилей)

ARMA – Autoregressive Moving Average (модель авторегресії та ковзного середнього)

ARIMA – Autoregressive Integrated Moving Average (інтегрована модель авторегресії та ковзного середнього)

SARIMA – Seasonal Autoregressive Integrated Moving Average (сезонна інтегрована модель авторегресії та ковзного середнього)

ВСТУП

Часовий ряд – це сукупність точок даних, індексованих (або перерахованих або схрещених) у часовому порядку. Найчастіше часовий ряд – це послідовність, яка взята в послідовно розташованих точках у часі. Таким чином, це послідовність даних дискретного часу. Прикладами часових рядів є висота океанських припливів, кількість сонячних плям та денна ціна закриття промислового середнього індексу Dow Jones.

Часові ряди дуже часто побудовані за допомогою лінійних діаграм. Часові ряди використовуються в статистиці, обробці сигналів, розпізнаванні шаблонів, економетриці, математичних фінансах, прогнозуванні погоди, прогнозуванні землетрусів, електроенцефалографії, управлінській техніці, астрономії, інженерії комунікацій і значною мірою в будь-якій галузі прикладної науки та техніки, що передбачає часові вимірювання.

Аналіз часових рядів включає методи аналізу даних часових рядів з метою отримання значущих статистичних даних та інших характеристик даних. Прогнозування часових рядів – це використання моделі для прогнозування майбутніх значень на основі раніше спостережуваних значень.

Це схоже на інші статистичні підходи до навчання, такі як навчання під наглядом або без нагляду. Однак прогнозування часових рядів має багато нюансів, які відрізняють його від звичайного машинного навчання. Багато компаній досліджують прогнозування часових рядів як спосіб прийняття кращих бізнес-рішень. Візьміть готель як приклад. Якщо менеджер має гарне уявлення про те, скільки гостей очікувати наступного літа, він може використовувати цю інформацію для планування управління персоналом, бюджету чи навіть розширення об'єктів. Так само впевнені уявлення про майбутні події можуть принести користь широкому спектру галузей та проблем, від традиційного сільського господарства до транспортування на замовлення тощо.

Існують дві основні мети аналізу часових рядів: визначення природи ряду і прогнозування (передбачення майбутніх значень часового ряду по теперішнім і минулим значенням). Обидві ці цілі вимагають, щоб модель ряду була ідентифікована і, більш-менш, формально описана. Як тільки модель визначена, ви можете з її допомогою інтерпретувати отримані дані (наприклад, використовувати у вашій теорії для розуміння сезонних змін цін на товари, якщо займаєтеся економікою). Не звертаючи уваги на глибину розуміння і справедливість теорії, ви можете екстраполювати потім ряд на основі знайденої моделі, тобто передбачити його майбутні значення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Часові ряди

Часовий ряд - це деяка кількість точок даних, упорядкованих за часом, це послідовність чітко визначених точок даних, виміряних через послідовні інтервали часу протягом певного періоду часу. Дані, зібрані на спеціальній основі або нерегулярно, не формують часовий ряд [1].

У часових рядах час часто є незалежною змінною, і мета зазвичай полягає в тому, щоб зробити прогноз на майбутнє.

У часових рядів є компоненти – тренд, сезонність, шум чи випадковість та рівень. Перш ніж перейти до визначення цих термінів, важливо зазначити, що не всі дані часових рядів включатимуть кожен із цих компонентів часових рядів. Наприклад, аудіофайли, які приймаються послідовно, є прикладами даних часових рядів, однак вони не містять сезонних компонентів (хоча зауважте, вони мали б періодичні цикли) [14 - 16]. З іншого боку, більшість бізнес-даних, ймовірно, містять сезонність, таку як роздрібні продажі, що досягають піку в четвертому кварталі [6].

Компоненти часового ряду:

– рівень: "рівень" або "індекс рівня" даних часових рядів – мається на увазі середнє значення;

– шум: дані всіх часових рядів матимуть шум або випадковість у точках даних, які не співвідносяться з поясненими тенденціями. Шум несистематичний і є короткочасним;

– сезонність: якщо в серії є регулярні та передбачувані коливання, які співвідносяться з календарем – це можуть бути щоквартальні, щотижневі або навіть дні тижня, то серія включає компонент сезонності. Важливо зауважити, що сезонність залежить від домену, наприклад, продажі нерухомості зазвичай вищі в літні місяці порівняно з зимовими місяцями, тоді як регулярний роздріб зазвичай досягає піку протягом кінця року. Крім того, не всі часові серії мають сезонний компонент, як, наприклад, аудіо та відео дані [13, 17];

– тренд (тенденція): коли йдеться про "тенденцію" даних часових рядів, це означає, що дані мають довгострокову траєкторію, яка може бути або в позитивному, або в негативному напрямку. Прикладом тенденції може бути довготривале збільшення даних по продажі;

– цикл: цикл часто описують як нефіксований паттерн, як правило, тривалістю принаймні 2 роки. Тривалість циклу описується як період. Прикладом даних часових

рядів, що демонструють циклічну поведінку, є збирання дичини або риби. Наприклад, збирання морського окуня з Банку Жоржа. Цикл може мати період тривалості від 4 до 5 років. Однак популяція морського окуня залежить від кількості відкладеної ікри(і зовнішнього впливу), для виживання [2].

На рисунку 1.1 продемонстровано основні компоненти часового ряду з продажу автомобілів у період з січня 2016 року по липень 2018. Верхній графік демонструє собою початкові дані – дані, які розглядаються для аналізу. Далі знаходиться графік тренду, який демонструє зростання продажів з роками. Після тренду продемонстрована сезонність, з якої можна зробити висновок, що зростання та спад продажів повторюється щороку в однакові місяці. Останній графік – шум.

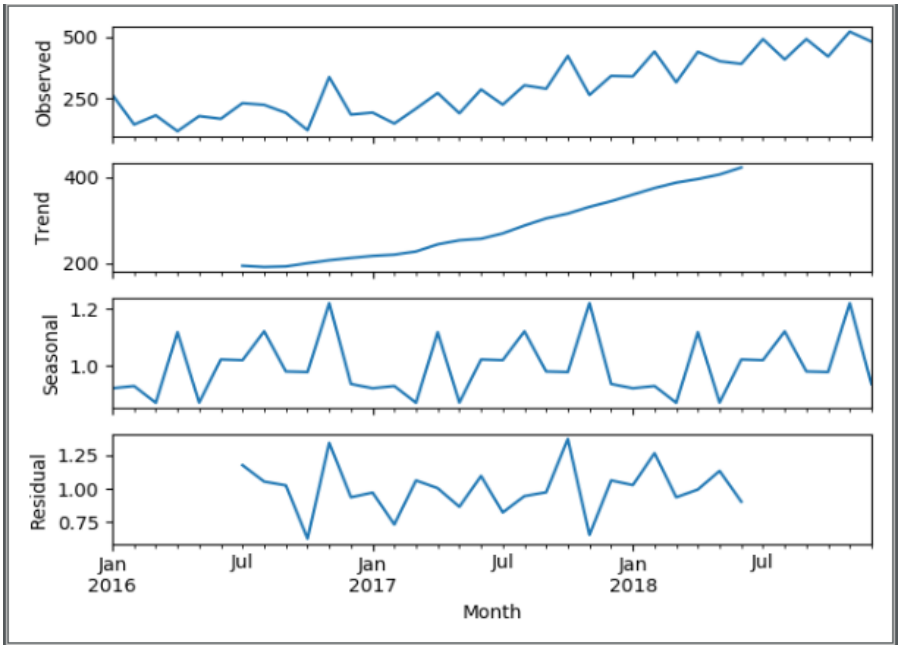


Рисунок 1.1 – Основні компоненти часового ряду

1.2 Аналіз часових рядів

Аналізом часових рядів є використання статистичних методів для аналізу даних часових рядів та отримання значущих статистичних даних та інших характеристик даних часових рядів. Аналіз фокусується на порівнянні значень одного часового ряду або декількох залежних часових рядів у різні моменти часу [42].

Аналізування допомагає нам зрозуміти, які показники приводять до певної тенденції у точках даних часових рядів, і допомагає нам у прогнозуванні та моніторингу точок даних, застосовуючи до них відповідні моделі для аналізу [30].

Для аналізу часових рядів існує багато методів, які досліджують компоненти часових рядів для виявлення закономірностей та додаткової інформації, які допоможуть у подальшій роботі з часовим рядом [26].

1.2.1 Аналіз тренду

Так як тренд це складова часових рядів, яка постійно повинна або зростати або зменшуватися (на невеликих відрізках можуть бути скачки, але на відстані тренд повинен дотримуватися або зростання або спаду), то провести аналіз тренду є не дуже складним завданням. Для цього можна використовувати згладжування.

Згладжування це процес який усереднює дані таким чином, щоб якісь несистематичні спостереження на проміжку часу були «вирівняні». Найпоширеніша техніка згладжування - ковзне середнє. Ковзне середнє виконує заміну елемента часового ряду на середнє або середнє зважене значення елементів, які знаходяться рядом, кількість сусідніх елементів визначається шириною інтервалу [46].

У разі, коли у вибірці даних є нерегулярні випадкові «викиди», рекомендується використовувати інший метод, так як «викиди» впливають середнє значення на інтервалі і застосування змінного середнього не є оптимальним рішенням.

Альтернативою використання середнього значення є використання медіани всіх елементів, які потрапили до інтервалу. Перевагою використання медіани є те, що вона менш чутлива до «викидів» - нехарактерним значенням вибірки, які є або занадто великими або маленькими.

Якщо ж помилка обчислення велика, рекомендується вдатися до зваженого методу найменших квадратів або методу негативного експоненційного зваженого згладжування. Перевагою використання цих методів є те, що вони фільтрують шуми і перетворюють дані в криву на яку не впливають «викиди».

Для часових рядів з малою кількістю даних може бути використаний метод бікубічних сплайнів [3].

1.2.2 Аналіз сезонності

Сезонність одна з основних компонентів часового ряду, яка зазвичай визначено як кореляційна залежність порядку k між кожним i-тим елементом ряду, та (i-k) елемент обчислюється за допомогою автокореляції (кореляція між 2 значеннями). Порядок автокореляції k називається лагом. У разі, якщо помилка вимірювання невелика, візуально ідентифікувати сезонність на графіку можна як повторення кривої прогнозу якогось візерунка кожні k елементів [25].

Дослідити сезонність можна за допомогою коррелограмми. Коррелограмми (автокоррелограмми) графічно відображають функцію автокореляції (autocorrelation function) ACF. ACF відображає кореляцію елементів і їх помилок для послідовності лагів з певного проміжку. Діапазон значень для кожного лага відображається на коррелограммі, але при аналізі сезонності прийнято вважати що розмір автокореляцій важливіший, ніж достовірність значень, так як аналізуються великі (значущі) значення автокореляції на лагах [5].

При аналізі коррелограмм слід пам'ятати, що автокореляції для послідовних лагів є залежними значеннями, наприклад, якщо перший елемент залежить від другого, а другий від третього, то ми можемо стверджувати, що перший елемент так само в деякій мірі залежить від третього. З цього можна зробити висновок, що при диференціації часового ряду по лагу 1 (вилучення автокореляції першого порядку) інші залежності можуть сильно змінитися.

Крім автокореляції існує так само метод часткової автокореляції (partial autocorrelation function) PACF. Метод, який є деяким розширенням методу автокореляції, в якому залежність проміжних елементів не враховуються. Іншими словами, PACF аналогічна ACF, за винятком того, що при розрахунку, кореляції між елементами всередині лага не враховуються.

Для виявлення прихованих властивостей ряду можна вдатися до видалення сезонної залежності. Для цього ряд необхідно продиференціювати. Результатом диференціації для лага k буде заміна кожного i елемента як різницю i-к елемента. Диференціація дозволяє прибрати деякі автокореляції, які, в свою чергу, впливають на інші автокореляції і це дозволяє проявити властивості сезонності, які до цього не були яскраво виражені. Крім того, диференціація дозволяє привести часовий ряд до стаціонарного виду, а це необхідно для побудови моделі прогнозування [5].

1.2.3 Експоненційне згладжування

Експоненційне згладжування даних часових рядів призначає експоненційне зменшення ваги від новіших до найстаріших спостережень. Іншими словами, чим старші дані, тим менша пріоритетність («вага») даних; новіші дані розглядаються як більш релевантні та мають більший пріоритет. Параметри згладжування (константи згладжування) - зазвичай позначаються як α - визначають пріоритетність для спостережень [10].

Експоненційне згладжування зазвичай використовується для складання короткострокових прогнозів, оскільки довгострокові прогнози за допомогою цієї методики можуть бути досить ненадійними.

Просте (одинарне) експоненційне згладжування використовує середньозважену ковзне середнє з експоненційне зменшенням ваги.

Подвійне експоненційне згладжування Холтом, скореговане трендом, зазвичай більш надійне для обробки даних, що показують тенденції, порівняно з простим експоненційним згладжуванням.

Потрійне експоненційне згладжування (його також називають мультиплікативними Холта-Вінтерса), як правило, більш надійне для параболічних трендів або даних, що показують тренд та сезонність.

1.2.3.1 Просте експоненційне згладжування

Простою і прагматичною моделлю [22] для часового ряду було б вважати кожне спостереження таким, що складається з константи та компонента помилки (epsilon), тобто:

$$X_t = b + \varepsilon_t .$$

Константа b є відносно стабільною у кожному сегменті ряду, але з часом може повільно змінюватися. Якщо можливо, то одним із способів виділити справжнє значення b, а отже, систематичну чи передбачувану частину ряду, є обчислення свого роду ковзного середнього, де поточним та попереднім ("молодші") спостереженням присвоюються більша вага ніж відповідні старіші спостереження. Просте експоненційне згладжування

здійснює саме таке зважування, де старішім спостереженням присвоюється експоненційно менша вага. Формула простого експоненційного згладжування:

$$S_t = \alpha X_t + (1 - \alpha) S_{t-1} .$$

При рекурсивному застосуванні до кожного послідовного спостереження в серії кожне нове згладжене значення (прогноз) обчислюється як зважене середнє значення поточного спостереження та попереднього згладженого спостереження; попереднє згладжене спостереження обчислювалось по черзі від попереднього спостережуваного значення та згладженого значення перед попереднім спостереженням тощо. Таким чином, фактично кожне згладжене значення є середньозваженим середнім показником попередніх спостережень, де ваги зменшуються експоненційно залежно від значення параметра (альфа). Якщо дорівнює 1 (одиниця), то попередні спостереження повністю ігноруються; якщо дорівнює 0 (нуль), то поточне спостереження повністю ігнорується, а згладжене значення повністю складається з попереднього згладженого значення (яке, в свою чергу, обчислюється з згладженого спостереження перед ним тощо), таким чином, всі згладжені значення будуть бути рівним початковому згладженому значенню S0). Значення між ними дають проміжні результати.

Для оцінки найкращого параметру альфа на практиці часто вибирається за допомогою сітки пошуку простору параметрів; тобто різні рішення спробують починати, наприклад, з 0,1 до 0,9, з кроком 0,1. Потім вибирається таким чином, щоб отримати найменші суми квадратів (або середні квадрати) для залишків (тобто спостережувані значення мінус прогнози на крок вперед).

Найпростіший спосіб оцінки точності прогнозів на основі конкретного значення - це просто побудувати спостережувані значення та прогнози на крок вперед у вигляді графіку. Цей графік також може включати залишки (різниця між спостереженням та побудованим прогнозом), так що частини де модель зробила кращий або гірший прогноз можна легко ідентифікувати відмінністю на графіку. Ця візуальна перевірка точності прогнозів найчастіше є найпотужнішим методом визначення того, чи підходить модель експоненційного згладжування для даних чи ні, але також існують спеціальні статистичні метрики [46], які дозволяють визначити оптимальне значення параметру альфа.

Середня помилка: середнє значення помилки (ME) просто обчислюється як середнє значення помилки (середнє спостережуване мінус прогноз на один крок вперед). Очевидно, недоліком цього заходу є те, що значення позитивних та негативних помилок

можуть скасовувати одне одного, тому цей захід не є дуже хорошим показником загальної придатності.

Середня абсолютна помилка: середнє значення абсолютної помилки (MAE) обчислюється як середнє значення абсолютної помилки. Якщо це значення дорівнює нулю, відповідність (прогноз) ідеальна. У порівнянні із середнім значенням помилки у квадраті, цей показник придатності буде «підкреслювати» залишків, тобто унікальні або рідкісні великі значення помилок впливатимуть на MAE менше, ніж значення методу середніх квадратів.

Сума помилки у квадраті (SSE), Середня помилка у квадраті. Ці значення обчислюються як сума (або середня величина) значень квадратичної помилки. Це найпоширеніший показник невідповідності в статистичних процедурах пристосування.

Відсоткова помилка (PE). Усі вищезазначені заходи залежать від фактичного значення помилки. Може здатися розумним швидше виразити невідповідність у відносному відхиленні прогнозів на крок вперед від спостережуваних значень, тобто відносно величини спостережуваних значень. Наприклад, намагаючись передбачити щомісячні продажі, які можуть коливатися (наприклад, сезонно) від місяця до місяця, ми можемо бути задоволені, якщо наш прогноз "потрапить у ціль" з точністю приблизно до 10%. Іншими словами, абсолютні помилки можуть викликати не стільки інтерес, скільки відносні помилки в прогнозах.

Середня процентна похибка (MPE). Це значення обчислюється як середнє значення PE.

Середня абсолютна процентна помилка (MAPE). Як і у випадку зі середнім значенням помилки (ME), середня відсоткова помилка, що становить близько нуля, може бути вироблена великими позитивними та негативними процентними помилками, які скасовують одна одну. Таким чином, кращою мірою відносного загального пристосування є середній абсолютний відсотковий похибку. Також ця міра зазвичай більш значуща, ніж середня помилка у квадраті. Наприклад, знання того, що середній прогноз "відхиляється" на ± 5%, є корисним результатом і сам по собі, тоді як середня квадратична помилка 30,8 не може бути відразу інтерпретована.

Крім простого експоненційного згладжування, були розроблені складніші моделі для аналізу часових рядів із сезонними та трендовими компонентами. Загальна ідея тут полягає в тому, що прогнози обчислюються не тільки за попередніми спостереженнями (як у простому експоненційному згладжуванні), але можна додати незалежну (згладжену) тенденцію та сезонну складову.

Багато даних часових рядів дотримуються періодичних сезонних моделей. Наприклад, щорічні продажі іграшок будуть, мабуть, піком у листопаді та грудні, а можливо, і влітку (зі значно меншим піком), коли діти перебувають на літніх канікулах. Ця закономірність, ймовірно, повторюватиметься щороку, однак відносна кількість приросту продажів протягом грудня може повільно змінюватися з року в рік. Таким чином, може бути корисним згладити сезонний компонент незалежно за допомогою додаткового параметра, який зазвичай позначається як дельта. Наприклад, протягом грудня місяця продаж певної іграшки може зростати на 1 мільйон доларів щороку. Таким чином, ми могли б додати до наших прогнозів на кожний грудень суму в 1 мільйон доларів (понад відповідну середньорічну кількість), щоб врахувати це сезонне коливання. У цьому випадку сезонність є адитивною. Або протягом грудня місяця продажі певної іграшки можуть зрости на 40%, тобто збільшитись в 1,4 рази. Таким чином, коли продажі на іграшку слабкі, то абсолютний приріст продажів протягом грудня буде відносно слабким (але відсоток буде постійним); якщо продажі іграшки сильні, то абсолютний приріст продажів буде пропорційно більшим. Знову ж таки, у цьому випадку продажі збільшуються за певним фактором, і, таким чином, сезонний компонент має мультиплікативний характер.

1.2.3.2 Подвійне експоненційне згладжування (метод Холта)

Подвійне експоненційне згладжування – це розширення до експоненційного згладжування, яке явно додає підтримку тенденціям у одномірному часовому ряду. На додаток до альфа-параметра для контролю коефіцієнта згладжування для рівня, додається додатковий коефіцієнт згладжування для контролю зменшення впливу зміни тенденції під назвою бета (b). Метод підтримує тенденції, які змінюються різними способами: адитивно та мультипликативно, залежно від того, лінійна чи експоненційна тенденція відповідно [46].

Адитивний тренд: подвійне експоненційне згладжування з лінійною тенденцією.

Мультиплікативний тренд: подвійне експоненційне згладжування з експоненційним трендом.

Подвійне експоненційне згладжування з адитивним трендом класично називається лінійною трендовою моделлю Холта, названої на честь розробника методу Чарльза Холта.

Цей метод включає рівняння прогнозу та два рівняння згладжування (одне для рівня та одне для тренду):

$$\text{Рівняння прогнозу: } \hat{y}_{t+h|t} = l_t + hb_t ,$$

$$\text{Рівняння рівня: } l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) ,$$

$$\text{Рівняння тренду: } b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} ,$$

де l_t позначає оцінку рівня ряду за часом t, b_t позначає оцінку тенденції (нахилу) рядів у часі t, α - параметр згладжування для рівня $0 \leq \alpha \leq 1$, та β^* параметр згладжування для тренду $0 \leq \beta^* \leq 1$.

Як і у випадку з простим експоненційним згладжуванням, рівняння рівня тут показує, що l_t середнє зважене спостереження y_t , і навчання прогнозу на один крок вперед t має вигляд $l_{t-1} + b_{t-1}$. Рівняння тренду показує що b_t середньозважене значення оціненої тенденції за часом t, на основі $l_t - l_{t-1}$, та b_{t-1} попередня оцінка тренду.

Функція прогнозування вже не є рівномірною, а тенденційною. Прогноз на h-кроків вперед дорівнює останньому оціненому рівню плюс h разів останнє оцінене значення тренду. Отже, прогнози є лінійною функцією h.

Прогнози, зроблені методом Холта, демонструють постійну тенденцію (зростає чи зменшується) на невизначений час у майбутнє. Емпіричні дані свідчать про те, що ці методи мають тенденцію до надмірного прогнозування, особливо для більш тривалих горизонтів прогнозу. Мотивовані цим спостереженням, Гарднер та Маккензі у 1985 ввели параметр, який на деякий час у майбутньому демпфує (“приглушує”) тенденцію до плоскої лінії. Методи, що включають демпфований тренд, виявилися дуже успішними, і, мабуть, найбільш популярними індивідуальними методами, коли прогнози необхідні автоматично для багатьох серій.

У поєднанні з параметрами згладжування α і β^* (зі значеннями від 0 до 1, як у методу Холта), цей метод також включає параметр демпфування $0 < \phi < 1$. Рівняння будуть мати наступний вигляд:

$$\hat{y}_{t+h|t} = l_t + (\phi + \phi^2 + \ldots + \phi^h)b_t;$$

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1});$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)\phi b_{t-1} .$$

Якщо ϕ=1 то метод ідентичний лінійному методу Холта. Для значень між 0 та 1 ϕ пригнічує тенденцію, так що вона наближається до деякої константи у майбутньому. Насправді прогнози сходяться до $l_T+\frac{\phi b_T}{(1-\phi)}$, де $h \to \infty$ для любого $0 < \phi < 1$. Це означає, що короткострокові прогнози мають тенденцію, тоді як довгострокові прогнози є постійними. На практиці ϕ рідко менше 0,8, оскільки демпфування дуже сильно впливає на менші значення. Значення ϕ близьке до 1 означатиме, що демпфовану модель не можна відрізнити від недемпфованої моделі. З цих причин ми зазвичай обмежуємось ϕ до мінімуму 0,8 і максимум 0,98.

1.2.3.4 Потрійне експоненційне згладжування (метод Холта–Вінтерса)

У 1960 році метод Холта був покращений одним з його студентів – Вінтерсом [23]. Вінтерс запропонував окрім тренду враховувати також і сезонність. Метод Холта–Вінтерса включає рівняння прогнозу та три рівняння згладжування - одне для рівня l_t, одне для тренду b_t та одне для сезонного компоненту s_t, з параметрами згладжування α, β*, та γ відповідно. Параметр m використовується для позначення частоти сезонності. Наприклад, для даних за квартал року m=4, для щомісячних m=12.

Існує дві варіанти цього методу, які відрізняються за характером сезонної складової [8]. Адитивний метод є кращим, коли сезонні зміни приблизно постійні через серію, тоді як мультиплікативний метод є кращим, коли сезонні зміни змінюються пропорційно рівню серії. При адитивному методі сезонна складова виражається в абсолютних виразах у масштабі спостережуваного ряду, а в рівнянні рівня ряд сезонно коригується відніманням сезонної складової. Протягом кожного року сезонна складова буде дорівнювати приблизно нулю. При мультиплікативному методі сезонний компонент виражається у відносному вираженні (відсотках), а ряд сезонно коригується шляхом поділу на сезонний компонент. Протягом кожного року сезонна складова становитиме приблизно m.

Для адитивного методу Холта–Вінтерса рівняння будуть мати вигляд:

$$\hat{Y}_{t+h|t}=l_t+hb_t+s_{t+h-m(k+1)};$$

$$l_t=\alpha\left(y_t-s_{t-m}\right)+(1-\alpha)(l_{t-1}+b_{t-1});$$

$$b_t=\beta^*(l_t-l_{t-1})+(1-\beta^*)b_{t-1};$$

$$s_t = \gamma\left(y_t - l_{t-1} - b_{t-1}\right) + (1-\gamma)s_{t-m},$$

де k є цілою частиною $\frac{(h-1)}{m}$, яка гарантує, що оцінки сезонних індексів, використані для прогнозування, походять з останнього року вибірки.

Рівняння рівня показує середньозважене середнє значення для сезонного спостереження $(y_t - s_{t-m})$ та для несезонного прогнозу $(l_{t-1} + b_{t-1})$ для часу t. Рівняння тренда є ідентичним лінійному методу Холта. Сезонне рівняння показує середньозважене значення між поточним сезонним індексом $(y_t - l_{t-1} - b_{t-1})$ та сезонним індексом цього ж сезону у попередньому році (m часових періодів тому).

Рівняння сезонного компоненту виглядає як

$$s_t = \gamma\left(y_t - l_{t-1} - b_{t-1}\right) + (1-\gamma^*)s_{t-m}.$$

Якщо ми замінимо l_t з рівняння згладжування рівня форми рівнянь вище, отримуємо

$$s_t = \gamma^*(1-\alpha)\left(y_t - l_{t-1} - b_{t-1}\right) + \left[1 - \gamma^*(1-\alpha)\right]s_{t-m},$$

що є ідентичним рівнянню згладжування для сезонного компонента, який ми тут вказуємо $\gamma = \gamma^*(1-\alpha)$. Звичайне обмеження параметра $0 \le \gamma^* \le 1$, що можна трактувати як $0 \le \gamma \le 1-\alpha$.

Форма рівнянь для мультиплікативного методу Холта-Вінтерса має наступний вигляд:

$$\hat{Y}_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)};$$

$$l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} - b_{t-1});$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1};$$

$$s_t = \gamma\frac{y_t}{(l_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m},$$

де α параметр згладжування даних, $0 < \alpha < 1$, β параметр згладжування тренду, $0 < \beta < 1$, та γ параметр згладжування сезонності, $0 < \gamma < 1$.

1.2.4 Аналіз розподілених лагів

Аналіз розподілених лагів – це спеціальний метод оцінки залежності між рядами з відставанням [18]. Наприклад, припустимо, ви робите комп'ютерні програми і хочете встановити залежність між числом запитів, що надійшли від покупців, і числом реальних замовлень. Ви могли б записувати ці дані щомісячно протягом року і потім розглянути залежність між двома змінними: число запитів і число замовлень залежить від запитів, але залежить з запізненням. Однак очевидно, що запитів буде більше ніж фактичних замовлень, тому можна очікувати, що кількість замовлень буде залежати від кількості запитів з деякою затримкою. Іншими словами, в залежності між числом запитів і числом продажів є часове зрушення (лаг).

У регресійному аналізі, що включає дані часових рядів, якщо модель регресії включає не тільки поточні, але й відсталі (минулі) значення пояснювальних змінних, її називають моделлю розподіленого відставання (лагів).

Якщо модель включає одне або більше відсталих значень залежної змінної серед її пояснювальних змінних, вона називається авторегресійною моделлю.

Таким чином,

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + u_t ,$$

являє собою модель розподілених лагів, тоді як

$$Y_t = \alpha + \beta X_t + \gamma Y_{t-1} + u_t ,$$

приклад авторегресійної моделі. Авторегресійні моделі також відомі як динамічні моделі, оскільки вони зображують шлях часу залежної змінної відносно її минулих значень.

Залежність змінної Y (залежної змінної) від іншої змінної X (пояснювальної змінної) рідко є миттєвим. Дуже часто Y відповідає на X з промігом часу. Такий проміжок часу і називається лагом.

1.2.4.1 Метод Койка

Койк запропонував метод оцінки моделей розподіленого відставання [18]. Припустимо, ми почнемо з нескінченної моделі розподіленого відставання:

$$Y_t = \alpha+\beta_0 X_t+\beta_1 X_{t-1}+\beta_2 X_{t-2}+\ldots+u_t \, .$$

Припускаючи, що β має однаковий знак, Койк припускає, що вони спадають геометрично:

$$\beta_k=\beta_0\lambda^k \, , \; k=0, 1, \ldots \, , \qquad (1.1)$$

де λ - ступінь зниження розподіленого лагу, що лежить на відрізку $0 < \lambda < 1$, та $1 - \lambda$ є швидкістю пристосування.

Запропонований метод припускає, що кожен наступний коефіцієнт β є чисельно менше ніж кожний попередній (це слідує з $\lambda < 1$), маючи на увазі, що по мірі відходу глибше у минуле, вплив цього лагу на Y_t стає значно меншим. Наприклад, можна обґрунтувати це тим, що очікується, що поточні та минулі доходи впливатимуть на витрати поточного споживання сильніше, ніж на доходи в далекому минулому. Геометрично схема Койка зображена на рисунку 1.2.

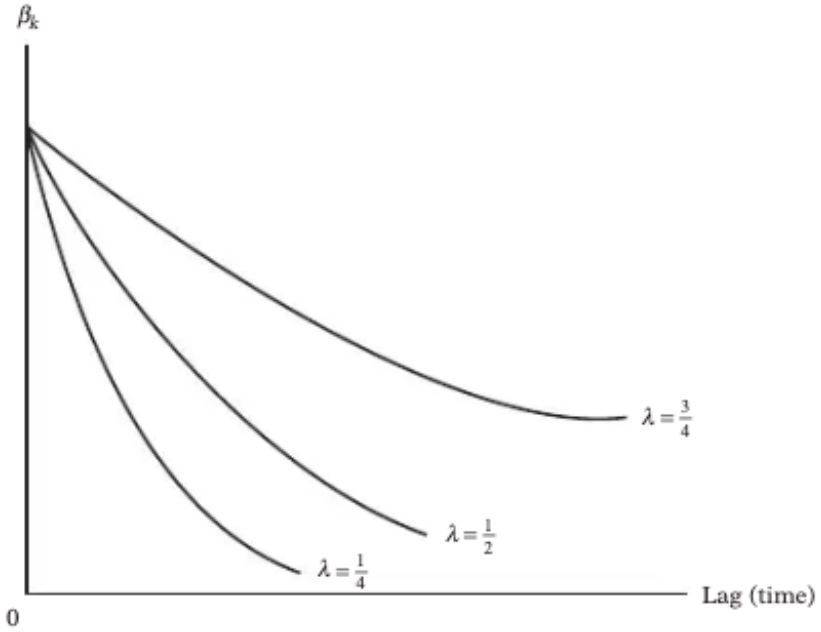


Рисунок 1.2 – Схема Койка

На рисунку можна відмити що значення β_k крім β_0 залежить також від λ. Чим ближче значення λ до 1 тим повільніше змінюється ступінь зниження β_k, тоді як чим ближче це значення до 0 тим швидше змінюється значення β_k. У першому випадку X має значущий вплив на Y_t, у той час як у другому випадку вплив на значення Y_t зменшується дуже швидко.

Також варто звернути увагу на деякі особливості моделі Койка:

– припускаючи що значення λ позитивні, Койк виключає можливість зміни знаку для β

– припускаючи що $\lambda<1$, значення β які являються новішими отримують більшу вагу, ніж β у минулому

– сума усіх β є кінцевою

$$\sum_{k=0}^{\infty} \beta_k = \beta_0 \left(\frac{1}{1-\lambda} \right) .$$

Як результат рівняння (1.1) модель безкінечного лагу матиме вигляд:

$$Y_t = \alpha + \beta_0 X_t + \beta_0 \lambda X_{t-1} + \beta_0 \lambda^2 X_{t-2} + \ldots + u_t . \qquad (1.2)$$

Але цю модель складно аналізувати через безкінечну кількість параметрів та через нелінійну форму параметру λ. Метод лінійного регресійного аналізу не може бути застосований до такої моделі.

Для вирішення цієї проблеми Койк запропонував зробити відставання від (2) на один період:

$$Y_{t-1} = \alpha + \beta_0 X_{t-1} + \beta_0 \lambda X_{t-2} + \beta_0 \lambda^2 X_{t-3} + \ldots + u_{t-1} \qquad (1.3)$$

помножити результат (1.3) на λ:

$$\lambda Y_{t-1} = \lambda\alpha + \lambda\beta_0 X_{t-1} + \beta_0 \lambda^2 X_{t-2} + \beta_0 \lambda^3 X_{t-3} + \ldots + \lambda u_{t-1} \qquad (1.4)$$

відняти (1.4) від (1.2)

$$Y_t - \lambda Y_{t-1} = \alpha(1 - \lambda) + \beta_0 X_t + (u_t - \lambda u_{t-1}) \quad (1.5)$$

спростивши (1.5) отримаємо

$$Y_t = \alpha(1 - \lambda) + \beta_0 X_t + \lambda Y_{t-1} + v_t ,$$

де $v_t = (u_t - \lambda u_{t-1})$ є ковзним середнім u_t та u_{t-1}.

Вищеописане і є перетворенням Койка, яке зводить процес оцінювання моделі розподілених лагів до знаходження значень тільки трьох невідомих: α, β_0 та λ, замість знаходження значень α та нескінченних значень β, як у звичайної моделі розподілених лагів

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \ldots + u_t .$$

1.2.4.2 Метод Алмона

Модель Койка широко використовується на практиці, але через те, що модель базується на припущенні що коефіцієнти β зменшується геометрично по мірі зростання відставання, у деяких випадках це припущення може бути занадто обмежуючим.

Наприклад, якщо у нас є лаги де значення змінної β спочатку зростають а потім зменшуються, або мають циклічний характер, то модель Койка тут буде неможливо використовувати.

Але можливо сприйняти β_i як функцію деякого і, що є довжиною лагу (часу), та побудувати криві, які відображатимуть функціональну залежність між ними, як продемонстровано на рисунку 1.3. Саме цю ідею і запропонував Ш. Алмон [18].

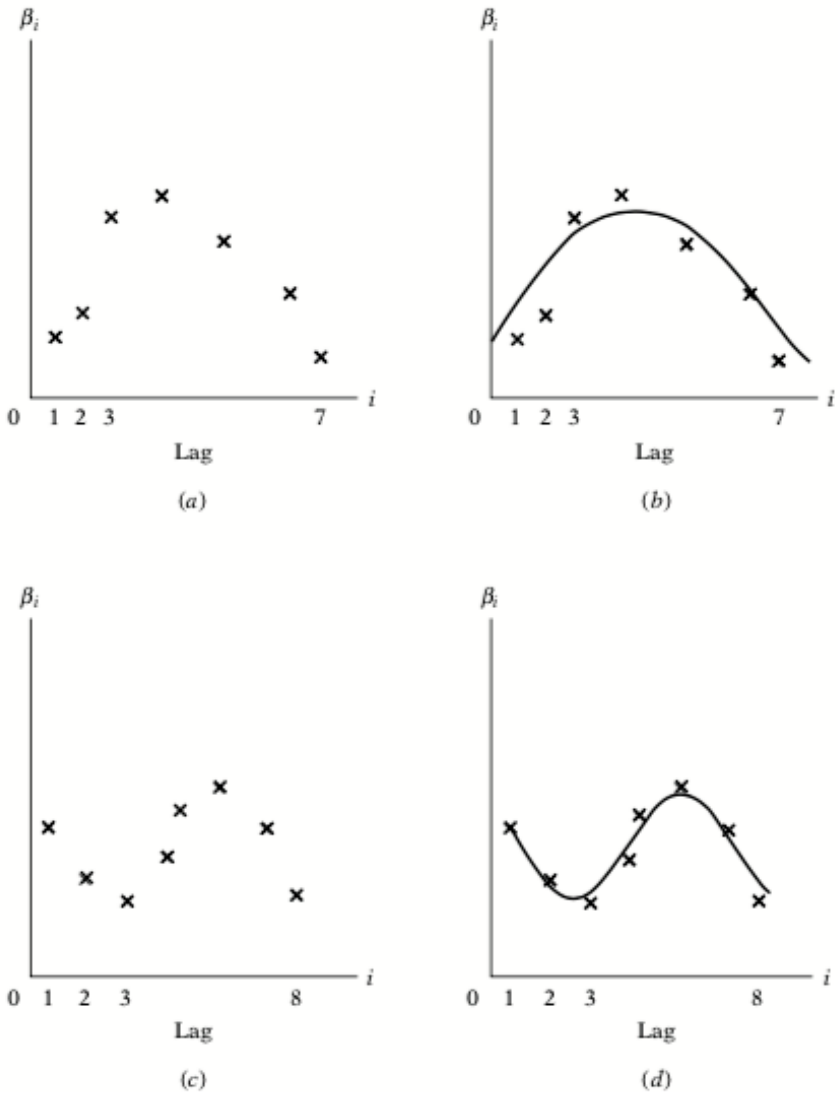


Рисунок 1.3 – Схема ідеї методу Алмона

Маючи кінцеву модель розподілених лагів, яка має вигляд:

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \ldots + \beta_k X_{t-k} + u_t ,$$

яка може бути записана як:

$$Y_t = \alpha + \sum_{i=0}^{k} \beta_i X_{t-i} + u_t \quad .$$

Слідуючи теоремі Вейєрштрасса, Алмон вважає що β_i можуть бути наближені поліномом відповідного ступеня у часі і, що є довжиною лагу.

Наприклад у ситуації (а) з рисунку 1.3 поліномом буде поліном другого ступеню з рівнянням

$$\beta_i = \alpha_0 + \alpha_1 i + \alpha_2 i^2 \ , \qquad (1.6)$$

у ситуації (c) з рисунку 1.3. поліном буде поліномом третього ступеню та матиме вигляд

$$\beta_i = \alpha_0 + \alpha_1 i + \alpha_2 i^2 + \alpha_3 i^3 \ .$$

У загальному випадку рівняння поліному буде мати вигляд

$$\beta_i = \alpha_0 + \alpha_1 i + \alpha_2 i^2 + ... + \alpha_m i^m \ ,$$

де m – ступінь поліному для часу i, k – максимальна довжина лагу. Вважається, що m повинна бути менше за k.

Для розгляду методу Алмона вважатимемо, що ми маємо поліном другого ступеня (1.6). Записавши його у вигляді

$$Y_t = \alpha + \sum_{i=0}^{k} \left(\alpha_0 + \alpha_1 i + \alpha_2 i^2\right) X_{t-i} + u_t \ =$$

$$= \alpha + \alpha_0 \sum_{i=0}^{k} X_{t-i} + \alpha_1 \sum_{i=0}^{k} i X_{t-i} + \alpha_2 \sum_{i=0}^{k} i^2 X_{t-i} + u_t \quad , \qquad (1.7)$$

де

$$Z_{0t} = \sum_{i=0}^{k} X_{t-i} \quad ,$$

$$Z_{1t} = \sum_{i=0}^{k} i X_{t-i} \quad ,$$

$$Z_{2t} = \sum_{i=0}^{k} i^2 X_{t-i} \quad .$$

Ми можемо переписати (1.7) як

$$Y_t = \alpha + \alpha_0 Z_{0t} + \alpha_1 Z_{1t} + \alpha_2 Z_{2t} + u_t \ .$$

У методі Алмона [18] значення Y регресує на перетворених значеннях Z, а не на вихідних значеннях X. Обчислення параметрів α та α_i отримане таким чином буде мати всі бажані статистичні властивості якщо компонент стохастичного збудження u задовольняє припущенням класичної лінійної регресійної моделі. У цьому відношенні

модель Алмона має значну перевагу над методом Койка, оскільки метод Койка має проблеми з обчисленням які є результатом наявності стохастичної пояснювальної змінної Y_{t-1} та її можливу кореляцію з компонентом похибки.

Після того, як були обчислені a, за допомогою нижче наведених формул можливо обчислити значення β:

$$\hat{\beta}_0 = \hat{a}_0;$$
$$\hat{\beta}_1 = \hat{a}_0 + \hat{a}_1 + \hat{a}_2;$$
$$\hat{\beta}_2 = \hat{a}_0 + 2\hat{a}_1 + 4\hat{a}_2;$$
$$\dots$$
$$\hat{\beta}_k = \hat{a}_0 + k\hat{a}_1 + k^2 \hat{a}_2 .$$

Встановивши значення m та k, можна записати рівняння для Z. Наприклад, якщо ступінь поліному m = 2, максимальна довжина лагу k = 5, то Z матиме вигляд:

$$Z_{0t} = \sum_{i=0}^{5} X_{t-i} = (X_t + X_{t-1} + X_{t-2} + X_{t-3} + X_{t-4} + X_{t-5}) ,$$
$$Z_{1t} = \sum_{i=0}^{5} iX_{t-i} = (X_{t-1} + 2X_{t-2} + 3X_{t-3} + 4X_{t-4} + 5X_{t-5}) ,$$
$$Z_{2t} = \sum_{i=0}^{5} i^2X_{t-i} = (X_{t-1} + 4X_{t-2} + 9X_{t-3} + 16X_{t-4} + 25X_{t-5}) .$$

Розібравши ці дві моделі можна зробити висновки, що метод Алмона більш гнучкий та дозволяє аналізувати різноманітні структури лагів, не тільки ті, де значення β зменшується геометрично, як у методі Койка. Також перевагою є те, що якщо ступінь поліному низький, то кількість коефіцієнтів a, які треба обчислити, значно менша від початкової кількості коефіцієнтів (β).

1.3 Сфери застосування часових рядів

Аналіз часових рядів використовується для визначення закономірностей, що існують у даних, щоб визначити модель, яка може бути використана для прогнозування майбутньої поведінки ділових показників [12, 41] (ціна фондового ринку, бюджет, продажі). Широким прикладом використання може бути середовище Інтернету речей (IoT), де дистанційно пристрої постійно фіксують показники для аналітичних цілей. Наприклад, моніторинг нафтових свердловин - це звичайний випадок використання IoT, коли аналіз численних показників з нафтової свердловини може допомогти передбачуваному технічному обслуговуванню, в якому аналіз може призвести до прогнозу коли обладнання вийде з ладу через тенденції та фактори, які представлені в даних. Також часові ряди можна використовувати для аналізу метрики комп'ютерної системи. У цій ситуації моніторинг показників комп'ютерних систем дозволяє IT-фахівцям контролювати стан різних систем. Такі показники, як використання пам'яті або кількість процесів, можна відслідковувати та оцінити, чи потрібно розгорнути нові комп'ютерні ресурси, чи потрібно перерозподілити програмні ресурси. Дані часових рядів можна використовувати для прогнозу прибутків чи аналізу продажів певних товарів у майбутньому. Існує багато областей де можна використати моделі часових рядів:

– бізнес: ланцюжок поставок, бронювання, веб-трафік;
– фінанси: біржовий варіант, біржа, економетрія;
– наука: астрономія, погода, прогноз землетрусів;
– техніка: датчики та управління обробкою сигналів;
– здоров’я: діагностика, біомедичний моніторинг.

Рисунок 1.4 – Приклад сфер де можна застосувати моделі часових рядів

1.4 Постановка задачі дослідження

Через швидкий розвиток інформаційних структур які доступні людині, кількість даних для опрацювання також збільшується. В багатьох сферах повсякденного життя можна застосовувати технології та обробляти отримані дані. Наприклад для прогнозу погоди, розрахування індексу Dow Jones чи прогнозу прибутків компанії. Ці масивні структури даних потребують належної обробки та ефективного аналізу. Через це задачу прогнозування на майбутнє можна вважати актуальною.

Метою роботи є дослідження результатів аналізу та прогнозування часових рядів. Враховуючи те, що існує багато методів для аналізу та прогнозування було прийняте рішення провести дослідження на декількох моделях та порівняти отримані результати прогнозів кожної із моделей. Дослідження буде проведено на декількох наборах даних. Перший набор даних з продажу автомобілів є більш різноманітним (містить інформацію про марку виробника, модель авто, рік виготовлення, місяць та рік продажу) та містить інформацію в період з 2007 року по 2017. Другий набір даних містить інформацію про продаж автомобілів за коротший проміжок часу - в період з 2016 по 2018 рік, має менше даних – місяць та рік продажу, кількість проданих автомобілів. Третій набір даних містить інформацію з 2015 по 2018 рік щобо росту цін на авокадо. Такі дані були обрані для того, щоб проаналізувати точність прогнозу на виборці даних з меншою кількістю записів методів прогнозування.

Результатом проведеного дослідження буде порівняння результатів прогнозування методами ARMA, ARIMA, SARIMA, Холта-Вінтерса (потрійне експоненційне згладжування) на різні проміжки часу (на рік, два роки). Для оцінки точності прогнозу буде використано методи MSE та RMSE, та графічний аналіз отриманих результатів на існуючих даних.

Також буде розроблено комп’ютерну програму для роботи з CRM системою, в якої користувач буде мати можливість самостійно підключатись до CRM системи та, обравши дані для аналізу зробити прогноз, або загрузити csv файл з комп’ютера.

Для дослідження алгоритмів прогнозування потрібно вирішити такі завдання:

- провести аналіз наборів даних;
- обрати набори даних на різний період часу;
- провести аналіз методів аналізу часових рядів;
- провести аналіз існуючих методів прогнозування;
- реалізувати алгоритм ARMA;

- реалізувати алгоритм ARIMA;
- реалізувати алгоритм SARIMA;
- реалізувати алгоритм Холта-Вінтерса;
- порівняти результати роботи алгоритмів на обраних наборах даних.

2 МОДЕЛІ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Загалом моделі даних часових рядів можуть мати багато форм і представляти різні стохастичні процеси. У літературі є дві широко використовувані лінійні моделі часових рядів – авторегресійні (AR) та моделі ковзного середнього (MA). Поєднуючи ці дві дві моделі, ми отримаємо модель авторегресії та ковзаючого середнього (ARMA). В літературі також запропоновано розширення моделі ARMA, яке зветься інтегрована модель авторегресії (autoregressive integrated moving average) чи скорочено ARIMA [24].

2.1 Модель авторегресії та ковзного середнього – ARMA

2.1.1 Авторегресійна модель AR(p)

Для початку розглянемо авторегресійну модель порядку p, позначення якої часто скорочують до AR(p). Авторегресійна модель [21] демонструє лінійну залежність значень часового ряду у даний момент від попередніх значень цього ряду. Термін «авторегресія» вказує, що це регресія змінної проти самої себе. Формула моделі AR(p) має вигляд

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \varphi_p y_{t-p} + \varepsilon_t \quad ,$$

де ε_t – шум у часі t, $\varphi_i (i = 1,2, ..., p)$ параметри моделі (коефіцієнти авторегресії), c – константа. Ціла константа p – порядок моделі. Часто для спрощення обчислень константу c приймають рівною 0.

Авторегресійні моделі (рис 2.1) надзвичайно гнучкі в обробці широкого спектру різних паттернів часових рядів. Приклади часових рядів на рисунку 2.1 демонструють моделі авторегресії [7] першого AR(1) та другого порядку AR(2). Зміна параметрів $\phi_i (i = 1,2, ..., p)$ призведе до зміни паттерну часового ряду, у той час як зміна ε_t призведе до зміни розміру часового ряду.

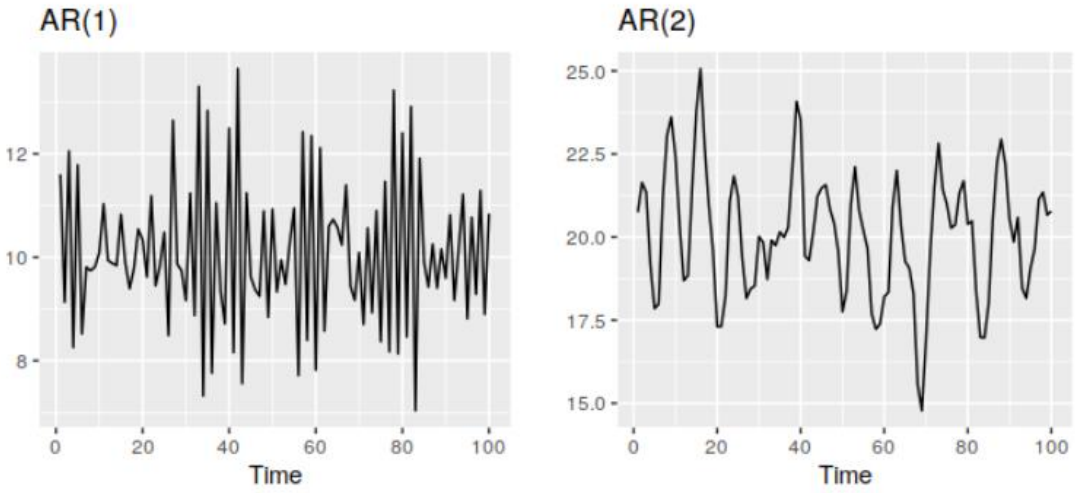


Рисунок 2.1 - Приклад моделей авторегресії першого та другого порядку

За допомогою методу Юла-Уолкера [32] можна провести оцінку підібраних параметрів. Існує пряма відповідність між ϕ_i(i = 1,2, ..., p) параметрами і коваріаційною функцією процесу, і цю відповідність можна перевернути для визначення параметрів функції автокореляції (яка сама отримується від коваріацій). Це робиться за допомогою рівнянь Юла-Уокера.

$$\gamma_m = \sum_{k=1}^{p} \varphi_k \gamma_{m-k} + \sigma_\varepsilon^2 \delta_{m,0} ,$$

де m=0, ... , p , дає $p + 1$, γ_m – автоковаріаційна функція, X_t, σ_ε - стандартне відхилення вхідного шуму та $\delta_{m,0}$ – дельта Кронекера. Враховуючи що остання частина рівняння не дорівнює нулю тільки якщо $m = 0$, то ці рівняння можна представити у вигляді матриці, де $m > 0$

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_p \end{bmatrix} = \begin{bmatrix} \gamma_0 & \gamma_{-1} & \gamma_{-2} & \cdots \\ \gamma_1 & \gamma_0 & \gamma_{-1} & \cdots \\ \gamma_2 & \gamma_1 & \gamma_0 & \vdots \\ \vdots & \vdots & \vdots & \cdots \\ \gamma_{p-1} & \gamma_{p-2} & \gamma_{p-3} & \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_p \end{bmatrix},$$

цю матрицю можна обчислити для всіх (φ_m; m = 1, 2, ... , p). Після обчислення матриці залишається рівняння для $m = 0$ яке має вигляд:

$$\gamma_m = \sum_{k=1}^{p} \varphi_k \gamma_{m-k} + \sigma_\varepsilon^2 ,$$

та знаючи $(\varphi_m; m = 1, 2, \dots , p)$ його можна обчислити для σ_ε^2.

2.1.2 Модель ковзного середнього MA(q)

Модель ковзного середнього (moving average, MA(q)) (рис 2.2) обумовлює лінійну залежність вихідного параметру від поточного та багатьох попередніх значень стохастичних умов. Замість використання минулих значень прогнозу регресійно, модель ковзного середнього [20] використовує минулі значення помилок прогнозу у моделі схожу на регресійну. Це означає, що MA(q) модель бачить випадковий шум безпосередньо на кожному поточному значенні моделі. На відміну від AR(p) моделі, яка бачить випадковий шум тільки опосередковано, шляхом регресії на попередні умови часового ряду. Ключова відмінність полягає в тому, що модель MA завжди буде бачити останні сплески шумів q для будь-якої конкретної моделі MA(q), тоді як модель AR(p) враховуватиме всі попередні сплески, хоча й у слабкій мірі.

$$y_t = c+\varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2}+ \dots + \theta_q\varepsilon_{t-q} ,$$

де $\theta_1, \dots, \theta_q$ – параметри моделі та $\varepsilon_t, \varepsilon_{t-1}, \dots \varepsilon_{t-q}$ – помилки шуму. Параметр q називається порядком моделі MA.

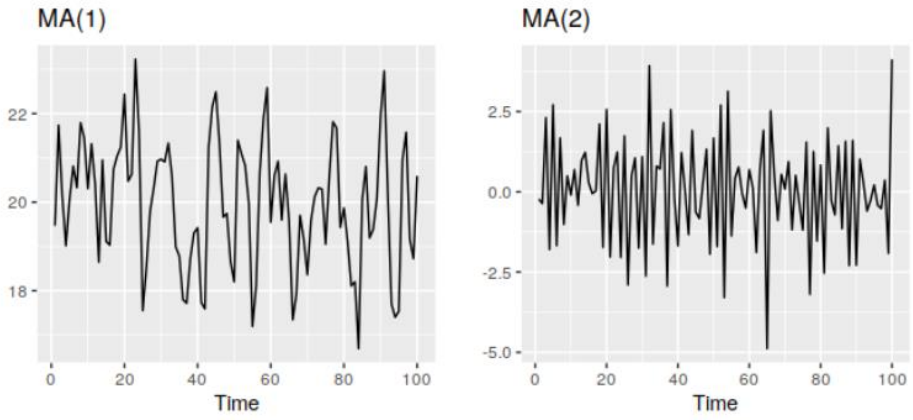


Рисунок 2.2 – Приклад моделей ковзного середнього першого та другого порядку

Змінюючи значення параметрів $\theta_1, ..., \theta_q$ ми отримаємо різні паттерни часового ряду. Так само як і у випадку з авторегресійною моделлю, змінюючи значення параметру помилки – ε_t, не змінить паттерн часового ряду, тільки його розмір.

Також будь-яку стаціонарну модель AR(p) можливо записати як MA(∞) модель. Використовуючи повторну заміну, ми можемо побачити це на прикладі моделі AR(1):

$$y_t=\phi_1 y_{t-1}+\varepsilon_t =$$
$$=\phi_1\left(\phi_1 y_{t-2}+\varepsilon_{t-1}\right)+\varepsilon_t=$$
$$= \phi_1^2 y_{t-2}+\phi_1 \varepsilon_{t-1}+\varepsilon_t=$$
$$= \phi_1^3 y_{t-3}+\phi_1^2 \varepsilon_{t-2}+\phi_1 \varepsilon_{t-1} +\varepsilon_t \ .$$

За умови $-1<\phi_1<1$ значення ϕ_1^k буде зменшуватись по міри того як значення k буде збільшуватись. Саме тому ми отримаємо

$$y_t = \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2}+ \phi_1^3 \varepsilon_{t-3}+ .... \quad ,$$

що є безкінечною моделлю ковзного середнього MA(∞).

Наклавши деякі обмеження на параметри MA ми можемо перетворити модель MA на зворотню (інвертовану), це означає що будь яку інвертовану модель MA(q) можливо записати як AR(∞). Наприклад в нас є процес MA(1)

$$y_t =\varepsilon_t + \theta_1 \varepsilon_{t-1} \ ,$$

у вигляді AR(∞) останню помилку можна записати як лінійну функцію поточних та минулих спостережень

$$\varepsilon_t= \sum_{j=0}^{\infty} (-\theta)^j y_{t-j} \ .$$

Коли $|\theta| > 1$ чим віддаленіше спостереження тим більший вплив воно має на помилку. Коли $|\theta| = 1$ віддалені спостереження мають такий саме вплив як і останнє спостереження. Оскільки жодна з цих ситуацій не є бажаним результатом ми ставимо як умову $|\theta| < 1$, тоді останні спостереження мають більшу вагу ніж спостереження із минулого. Таким чином процес є інвертованим за умовою $|\theta| < 1$.

Авторегресійна модель AR(p) та модель ковзного середнього MA(q) можуть бути поєднані для формування загального класу моделей часових рядів, відомих як моделі ARMA. Важливість цих моделей полягає у тому, що

вони роблять можливим моделювання для більш широкого спектру залежних структур та вони більш економні – дуже часто моделі ARMA(p,q) вимагають менше параметрів ніж чисті моделі AR(p) та MA(q). Модель ARMA(p, q), яка може використовуватись для одновимірного моделювання часових рядів, математично представлена як

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \ .$$

Змінні p та q посилаються на p – коефіцієнт авторегресії та q – коефіцієнт ковзного середнього.

Зазвичай моделями ARMA можна керувати за допомогою оператора відставання [18]. Оператор відставання (лагу) представлений у вигляді

$$Ly_t = y_{t-1} \ .$$

Поліноми оператора відставання (лаговий оператор) або поліноми відставання (лагу) для моделей ARMA представлені наступним чином:

$$\text{AR(p): } \varepsilon_t = \varphi(L) y_t \ ,$$
$$\text{MA(q): } y_t = \theta(L) \varepsilon_t \ ,$$
$$\text{ARMA(p, q) } = \varphi(L) y_t = \theta(L) \varepsilon_t \ ,$$

де $\varphi(L) = 1 - \sum_{i=1}^{p} \varphi_i L^i$ та $\theta(L) = 1 + \sum_{j=1}^{q} \theta_j L_j$.

Процес AR(p) записаний у вигляді $\varepsilon_t = \varphi(L) y_t$, його характеристичне рівняння матиме вигляд $\varphi(L) = 0$. Як було відмічено Боксом та Дженкінсом, наявність усіх коренів характеристичного рівняння поза значенням 1 є необхідною умовою для стаціонарності процесу AR(p). Наприклад, модель першого порядку AR(1)

$$y_t = c + \varphi_1 y_{t-1} + \varepsilon_t$$

буде стаціонарною за умовою

$$|\varphi_q| < 1$$

з константою

$$\mu = \frac{c}{1-\varphi_1}$$

та константною дисперсією

$$\gamma_0 = \frac{\sigma^2}{1-\varphi_1} \quad .$$

Процес MA(q) завжди є стаціонарним незалежно від значень параметрів MA. Умови щодо стаціонарності та інвертованості процесів AR та MA також стосуються процесу ARMA [43].

Процес ARMA(p,q) є стаціонарним якщо всі корені характеристичного рівняння $\varphi(L) = 0$ лежать за межами 1. Так само, якщо усі корені рівняння $\theta(L) = 0$ лежать за межами 1, процес ARMA(p,q) є інвертованим та може бути записаний як AR(∞). Через це на практиці можна зустріти випадки підбору параметрів моделі ARMA(p,q) шляхом заміни на підбір параметрів до моделі AR чи MA [36].

На рисунку 2.3 продемонстровано 2 графіки – функції автокореляції (autocorrelation function) ACF та функції часткової автокореляції (partial autocorrelation function) PACF [44], які використовуються для визначення значень AR(p) та MA(q). ACF діаграма - діаграма коефіцієнтів кореляції між часовим рядом та його лагів. PACF діаграма - діаграма часткової кореляції між спостереженням k періодів назад та поточним спостереженням, не враховуючи спостереження на проміжних лагах.

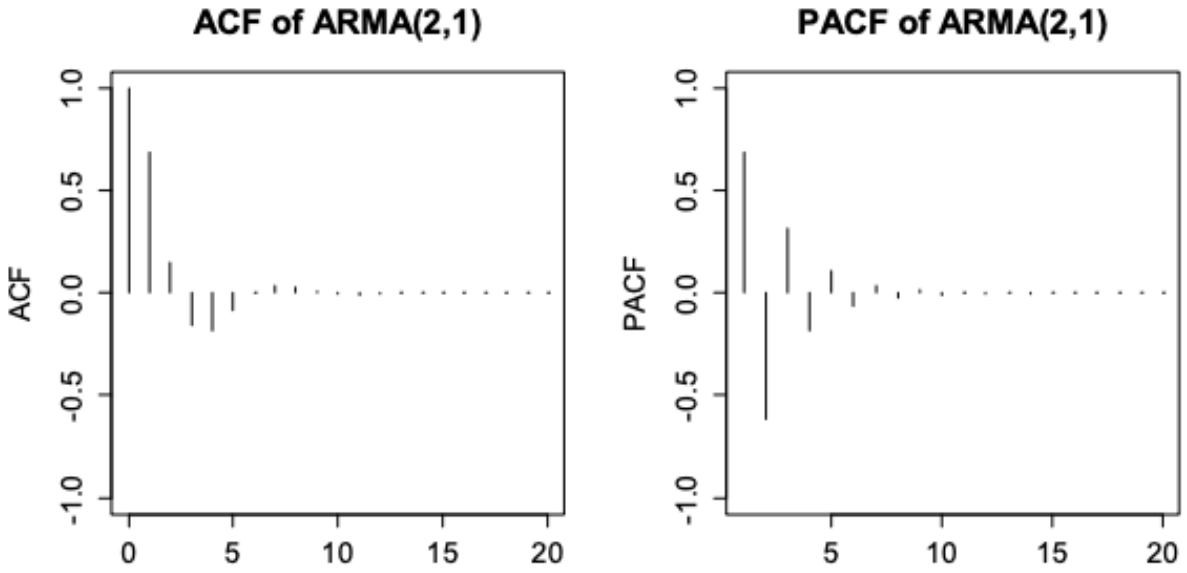


Рисунок 2.3 – Приклад підбору значень p та q

Для визначення належної моделі для часових рядів необхідно провести аналіз ACF та PACF. Ці статистичні графіки відображають, як спостереження у часових рядах пов'язані між собою. Для моделювання та прогнозування часто корисно побудувати графік ACF та PACF на основі послідовних часових лагів.

Рівняння ACF між двома змінними X_{t+k} та X_t представлено нижче

$$\mathrm{Corr}(X_{t+k}, X_t) = \frac{\mathrm{Cov}(X_{t+k}, X_t)}{\sqrt{\mathrm{Var}(X_{t+k})\mathrm{Var}(X_t)}} .$$

Оскільки для стаціонарних рядів необхідною умовою є те, що моменти не повинні змінюватись у часі, ми можемо виключити індекс t та записати автокореляцію лагу k у вигляді

$$\rho_k = \mathrm{Corr}(X_{t-k}, X_t) ,$$

де k = (0, 1, 2…).

Для оцінки кореляції спостережень найчастіше використовують принцип підставлення (plug-in estimation), який використовує оцінку автоковаріації як основу [45].

Дисперсія вибірки випадкової величини демонструє два аспекти зміщення оцінювача: по-перше, простий оцінювач є зміщеним, що може бути виправлено масштабуючими множниками; по-друге, незміщений оцінювач не є оптимальним з точки зору середньої квадратичної помилки (MSE), яку можна мінімізувати, використовуючи

інші масштабуючі множники, що призводить до зміщуваного оцінювача з меншим значенням MSE, ніж незміщений оцінювач. Простий оцінювач підсумовує квадратичні відхилення і ділиться на n, що є зміщенням. У той час як розділення на $n-1$ дасть об'єктивний (незміщенний) оцінювач.

MSE можна мінімізувати діленням на інше число (залежно від розподілу), але це призводить до зміщеного оцінювача. Число розподілу завжди більше n - 1, це називається оцінювач усадки, оскільки він «скорочує» незміщений оцінювач до нуля; для нормального розподілу оптимальне значення n + 1.

Припустимо, що ми маємо середню вибірку та некоректовану дисперсію вибірки

$$\overline{X}= \frac{1}{n}\sum_{t=1}^{n} X_t ; \qquad \hat{S}(k) = \sum_{t=1}^{n-k} (X_t - \overline{X})(X_{t+k} - \overline{X}) .$$

Зауважимо, що тут n використовується як знаменник незалежно від лагу i, отже, кількості доданків суми.

За допомогою функції автокореляції [35] ми спрощуємо аналіз, визначивши

$$\Pi_{n,k}= \frac{1}{n(n-2k)} \sum_{i=1}^{n} \sum_{j=k+1}^{n-k} \gamma|i - j| \quad .$$

Ця формула є середнім значенням кореляції у кореляційній матриці $n * n$, де кожна сторона усічена на k. Враховуючи стаціонарний процес з фіксованою середньою вибіркою та дисперсією вибірки ми маємо наступне рівняння коваріації

$$Cov(X_i, X_j) = \sigma^2\gamma|i-j| ,$$

з цієї формули виходить, що

$$E(X_iX_j) = \sigma^2\gamma|i-j| + \mu^2 .$$

Тоді

$$E[\hat{S}(k)] = E\left[\sum_{t=1}^{n-k} (X_t - \overline{X})(X_{t+k} - \overline{X})\right] =$$

$$=E\left[\sum_{t=1}^{n-k} X_t X_{t+k} - \bar{X}\left(\sum_{t=1}^{n-k} X_t + \sum_{t=1}^{n-k} X_{t+k}\right) + (n-k)\bar{X}^2\right] =$$

$$= E\left[\sum_{t=1}^{n-k} X_t X_{t+k} - \bar{X}\left(nX_t + \sum_{t=1}^{n-k} X_t\right) + (n-k)\bar{X}^2\right] =$$

$$=E\left[\sum_{t=1}^{n-k} X_t X_{t+k} - \bar{X}\sum_{t=k+1}^{n-k} X_t + k\bar{X}^2\right] =$$

$$=E\left[\left(\sum_{t=1}^{n-k} X_t X_{t+k}\right) - E\left(\bar{X}\sum_{t=k+1}^{n-k} X_t\right) - kE(\bar{X}^2)\right] =$$

$$=(n-k)(\sigma^2\gamma(k) + \mu^2) - (n-2k)(\sigma^2\Pi_{n,k} + \mu^2) - k(\sigma^2\Pi_{n,0} + \mu^2) =$$

$$=(n-k)\sigma^2\gamma(k) - \mu^2((n-k)\Pi_{n,k} + k(\Pi_{n,0} - \Pi_{n,k})) =$$

$$= (n-k)\sigma^2\left[\gamma(k) - \left(\Pi_{n,k} + \frac{k}{n-k}(\Pi_{n,0} - \Pi_{n,k})\right)\right].$$

Враховуючи оцінювач коваріації

$$\widehat{C}(k) = \frac{1}{n-k}\sum_{t=1}^{n-k}(X_t - \bar{X})(X_{t+k} - \bar{X}),$$

та підставивши це в рівняння вище, отримаємо

$$\frac{E(\widehat{C}(k))}{\sigma^2} = \gamma(k) - \left(\Pi_{n,k} + \frac{k}{n-k}(\Pi_{n,0} - \Pi_{n,k})\right).$$

Ця формула показує, що в нашому аналізі є зміщення. Однак для більшості стаціонарних процесів автокореляція розсіюється, оскільки спостереження з часом стають все далі один від одного. Це означає, що коли n велике, середні значення автокореляції $\Pi_{n,k}$ стають малими, і тому термін зміщення (другий доданок у виразі) також стає малим.

Коли дані мають тренд, автокореляція для малих лагів, як правило, є великою та позитивною, оскільки спостереження, що знаходяться поблизу мають майже однаковий розмір. З цього можна зробити висновок, що ACF для часового ряду з трендом буде мати позитивні значення, які будуть зменшуватись по мірі збільшення лагів. Якщо дані мають сезонність, ACF буде мати великі «сплески» тільки на сезонних лагах (при кратності сезонної частоти). Якщо дані мають і тренд і сезонність на графіку буде комбінація цих властивостей. Приклади наведені на рисунках 2.4, 2.5 та 2.6.

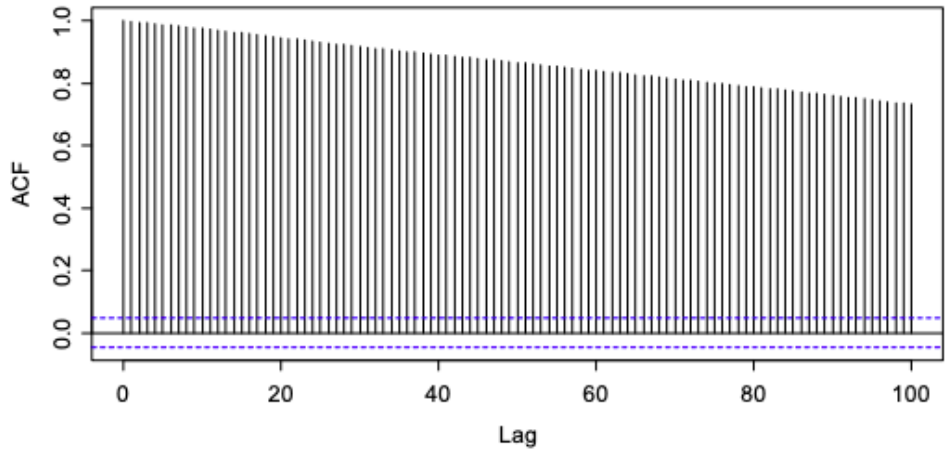


Рисунок 2.4 - Коррелограмма ACF з трендом

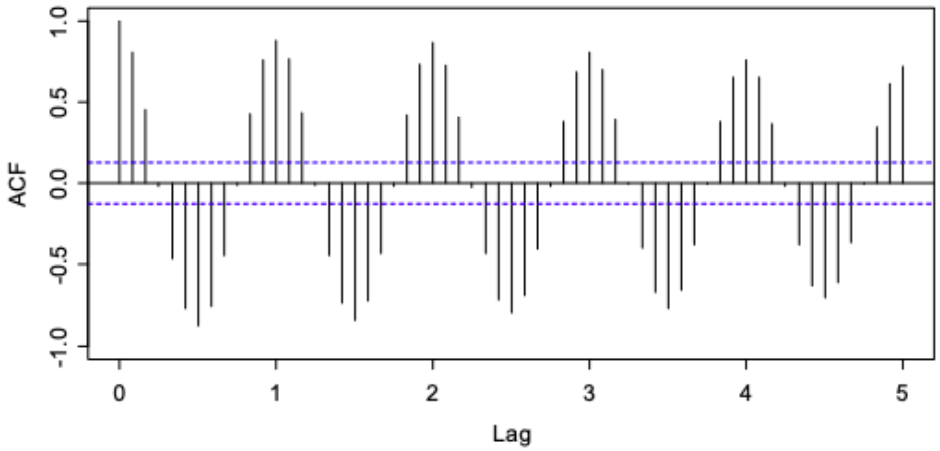


Рисунок 2.5 - Коррелограмма ACF з сезонністю

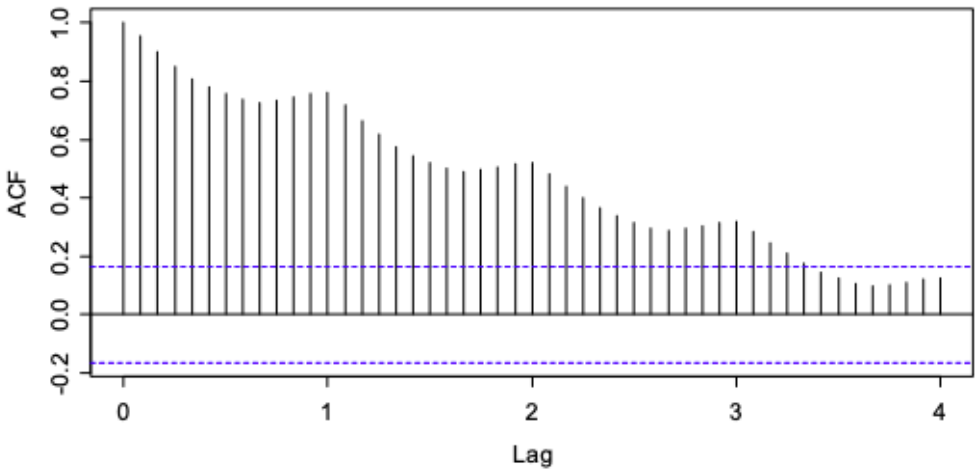


Рисунок 2.6 – Коррелограмма ACF з трендом та сезонністю

Часткова автокореляція – кореляція між спостереженнями часового ряду враховуючи проміжні спостереження. Наприклад, у нас є y – результуюча змінна, та x_1, x_2, x_3 – прогнозовані змінні (параметри рівняння регресії). Часткової кореляцією між y та x_3 є кореляція між змінними з урахуванням як y та x_3 пов'язані з x_1, x_2. При регресії PACF можна розрахувати шляхом кореляції залишків з двох регресій:

– регресія для прогнозу y з x_1, x_2;

– регресія для прогнозу x_3 з x_1, x_2;

– фактично виконується кореляція частин y та x_3 які не прогнозуються з x_1, x_2.

Рівняння для даного прикладу наведено нижче

$$\mathrm{PACF}(y_i,k) = \frac{\mathrm{Cov}(y\ , x_3|\ x_1,\ \ x_2)}{\sqrt{\mathrm{Var}(y|\ x_1, x_2)\ \mathrm{Var}(x_3|\ x_1, x_2)}}.$$

У загальному випадку рівняння PACF порядку k представлено у вигляді

$$\mathrm{PACF}(y_i,k) = \frac{\mathrm{Cov}(\ x_t\ , x_{t-k}|\ x_{(t-1)},\ \ x_{(t-2)}\ \ \ldots\ \ x_{(t-k+1)})}{\sqrt{\mathrm{Var}(x_t|\ x_{t-1}, x_{t-2},\ \ \ldots\ \ ,\ \ x_{(t-k+1)})\ \mathrm{Var}(x_{t-k}|\ x_{t-1}, x_{t-2}, ..x_{(t-k+1)})}}.$$

Характерно, що ні в ACF, ні в PACF значення лагу не відсікається строго на певному лагу. Натомість обидва графіки демонструють деяку нескінченну поведінку, наприклад, експоненційне зменшення у величині коефіцієнтів. Однак варте зазначити, що можливо помітити значний спад значень після якогось лагу. Якщо проаналізувати рисунок 2.3 можна помітити на графіку ACF, що після лагу 1 значення автокореляції значно зменшилося. З цього можна зробити висновок, що порядок моделі ковзного середнього MA(q) = 1. На графіку PACF характерний спад значень часткової автокореляції здійснюється після лагу 2, що свідчить про те, що порядок авторегресійної моделі AR(p) = 2.

2.2 Інтегрована модель авторегресії та ковзного середнього ARIMA(p,d,q)

Інтегрована модель авторегресії та ковзного середнього (autoregressive integrated moving average) ARIMA [11], іноді у літературі можна зустріти як модель Бокса – Дженкінса. Модель ARIMA є вдосконаленою версією моделі ARMA [31]. Вона дозволяє

зробити часовий ряд який має тренд стаціонарним за допомогою диференціювання, яке може бути застосовано на даних часового ряду один раз чи більше доки не буде виконана умова стаціонарності.

Ця абревіатура є описовою, відображаючи ключові аспекти самої моделі:

– AR (autoregression) – модель, яка використовує залежність між спостереженням та деякою кількістю відстаючих спостережень;

– I (integrated) – використання диференціації початкових даних (наприклад, віднімання спостереження від спостереження на попередньому етапі часу) для того, щоб зробити часовий ряд стаціонарним;

– MA (moving average) – модель, яка використовує залежність між спостереженням та помилкою.

Ці аспекти прямо задані у моделі як параметри. Стандартний запис моделі – ARIMA(p,d,q) [34], де параметри замінені на константи для швидкого запису конкретної моделі, яка є частиною ARIMA. Константи-параметри визначені як:

– p – порядок лагу (відставання) - кількість лагів спостережень, що входять у модель;

– d – ступінь диференціації - кількість разів початкові дані були продиференційовані;

– q – порядок ковзного середнього.

Формула моделі ARIMA(p,d,q) має наступний вигляд

$$y'_t = c + \varphi_1 y'_{t-1} + \ldots + \varphi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t \ ,$$

де y'_t – продиференційований часовий ряд, а доданки справа – параметри моделей AR та MA.

Як було зазначено, аналізувати та прогнозувати краще стаціонарні часові ряди, та модель ARIMA [19] за допомогою параметру d, який відповідає за диференціацію, приводить часові ряди до стаціонарності. Часовий ряд називається стаціонарним (рис. 2.7) якщо його властивості не залежать від часу у який проводиться аналіз. З визначення стаціонарного часового ряду можна стверджувати, що часові ряди які мають тренд чи сезонність не є стаціонарними, через те, що спостереження у різні моменти часу за наявності тренду чи сезонності впливають на отриманий результат. Деякі випадки можуть бути складнішими - часовий ряд із циклічною поведінкою (але без тренду чи сезонності) є стаціонарним. Це тому, що цикли не мають фіксованої довжини.

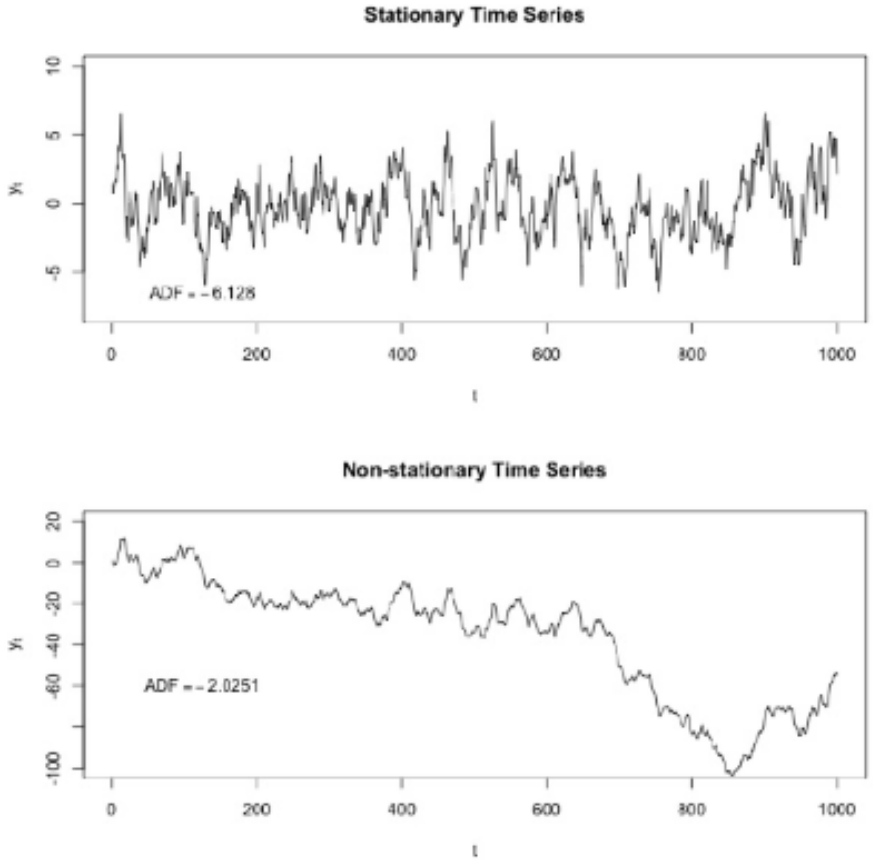


Рисунок 2.7 – Приклад стаціонарного та нестаціонарного часового ряду

Оцінити стаціонарність часового ряду можливо через графік даних ACF. Для стаціонарного часового ряду ACF (рис. 2.8) скоротиться до нуля порівняно швидко, тоді як ACF нестаціонарних даних буде зменшуватися повільно [29].

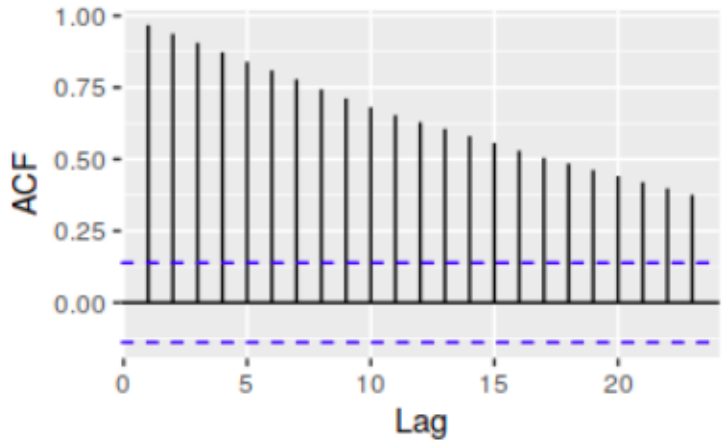


Рисунок 2.8 – ACF для нестаціонарного часового ряду

За допомогою логарифмування можна стабілізувати дисперсію часового ряду. Також використовуючи диференціацію можна допомогти стабілізувати середнє значення часового ряду, усунувши зміни рівня часового ряду, а отже, усунувши (або зменшивши) тренд та сезонність.

Продиференційований часовий ряд - це різниця між кожним послідовним спостереженням у початковому ряді, його можна записати наступним чином

$$y'_t = y_t - y_{t-1} \ .$$

Часовий ряд на якому виконали операцію диференціювання буде мати Т-1 значень, через те, що неможливо виконати диференціювання y'_t першого спостереження.

Якщо продиференційований часовий ряд є шумом, його рівняння матиме вигляд

$$y_t - y_{t-1} = \varepsilon_t \ ,$$

де ε_t – шум. Зробивши перестановку змінних ми отримаємо рівняння моделі випадкового блукання

$$y_t = y_{t-1} + \varepsilon_t \ .$$

Моделі випадкового блукання широко використовуються для нестаціонарних даних [39], зокрема фінансових та економічних даних. Вони зазвичай мають:

– тривалі періоди тренду вгору або вниз;

– раптові непередбачувані зміни напряму.

Прогнози за моделлю блукання дорівнюють останньому спостереженню, оскільки майбутні рухи непередбачувані і з однаковою ймовірністю будуть продовжуватись вгору чи вниз.

$$y_t = c + y_{t-1} + \varepsilon_t \ ; \ y_t - y_{t-1} = c + \varepsilon_t \ ,$$

де c – середнє значення різниці між послідовними спостереженнями. Якщо c – позитивне, значення y_t буде змінюватись вгору. В іншому випадку y_t буде змінюватись вниз.

Може виникнути ситуація, коли часовий ряд було продиференційовано, але ця операція не призвела до стаціонарності. У цьому випадку потрібно виконати диференціацію ще раз.

$$y''_t = y'_t - y'_{t-1} =$$
$$= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) =$$
$$= y_t - 2\,y_{t-1} + y_{t-2}\ .$$

За цих умов y''_t буде мати Т-2 значень. На практиці майже завжди достатньо 2 рівня диференціації.

Крім простої диференціації існує також сезонна диференціація – коли розраховується різниця між спостереженнями з одного сезону.

$$y'_m = y_t - y_{t-m}\ ,$$

де m – кількість сезонів.

На рисунку 2.9 продемонстровано як логарифмування стабілізує дисперсію часового ряду, а диференціація приводить його до стаціонарного вигляду.

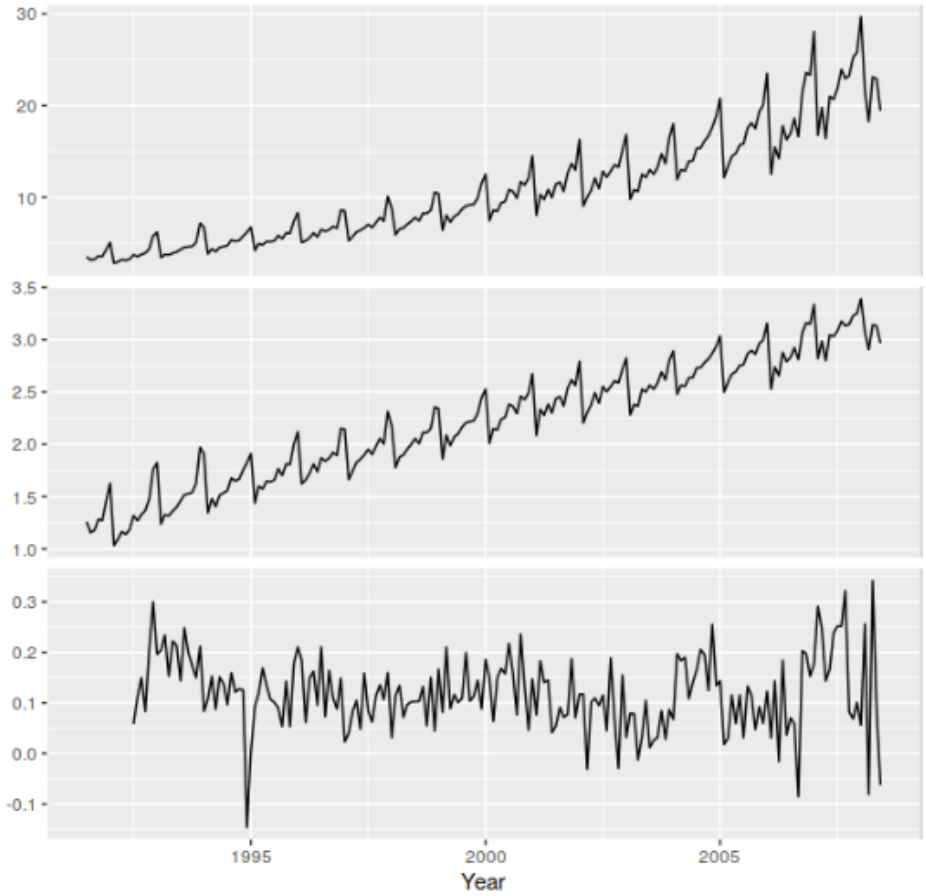


Рисунок 2.9 – Початкові дані (верхній графік), операція логарифмування (середній графік), сезонна диференціація (нижній графік)

Крім аналізу графіку ACF перевірити часовий ряд на стаціонарність можливо за допомогою тесту Діккі-Фуллера (Dickey – Fuller; DF). Тест Діккі – Фуллера перевіряє нульову гіпотезу про наявність одиничного кореня в авторегресивній моделі, тобто нестаціонарність ряду [4]. Альтернативна гіпотеза різна в залежності від того, яка версія тесту використовується, але зазвичай це стаціонарність (всі корені характеристичного поліному лежать поза значенням 1) або стаціонарність за умовою виключення тренду. Для великих та складних структур часових рядів, де потрібно враховувати процес авторегресії не тільки першого порядку рекомендується використовувати розширений тест Діккі-Фуллера (augmented Dickey-Fuller; ADF). Він названий на честь статистиків Девіда Діккі та Уейна Фуллера, які розробили тест у 1979 році. Існує три версії тесту DF, кожна з яких має свої критичні значення, котрі можна отримати з таблиці Діккі-Фуллера.

Тест DF перевіряє чи $\gamma = 0$ у моделі даних без константи та тренду

$$\Delta y_{t-1} = \gamma y_{t-1} + \varepsilon_t \ .$$

Якщо $\gamma = 1$, такий ряд не є стаціонарним та потребує приведення до стаціонарності.

Додавши до моделі вище вільний член ми отримаємо модель з константою але без тренду

$$\Delta y_{t-1} = \alpha + \gamma y_{t-1} + \varepsilon_t .$$

Також можна додати лінійний тренд, тоді модель буде мати вигляд

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \varepsilon_t ,$$

яку можна переписати у вигляді

$$\Delta y_t = y_t - y_{t-1} = \alpha + \beta t + \gamma y_{t-1} + \varepsilon_t ,$$

де y_t – початкові дані. Такий запис у формі регресійної моделі дозволяє нам провести лінійну регресію Δy_t та перевірити гіпотезу чи $\gamma = 0$.

У випадках, коли процес, який аналізується, є процесом не першого порядку рекомендується враховувати лаги перших різниць, а для аналізу таких даних треба застосовувати розширений метод Діккі-Фуллера (ADF).

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t ,$$

де α – константа, β – коефіцієнт тренду, p – порядок лагів авторегресійної моделі [27].

Тест ADF також має варіації. Виключивши з рівняння константу та коефіцієнт тренду ми отримаємо рівняння аналогічне випадковому блуканню (без константи та тренду)

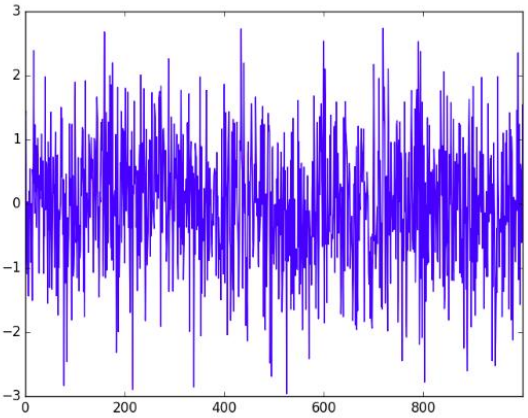
$$\Delta y_t = \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t .$$

Виключивши з рівняння лише коефіцієнт тренду ми отримаємо випадкове блукання з константою

$$\Delta y_t = \alpha + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t .$$

Як і звичайний тест DF, розширений перевіряє виконання умови стаціонарності ряду $\gamma = 0$.

На рисунку 2.10 нижче наведено приклади аналізу даних. Зверху зображено графік шуму, тому що ми напевно знаємо, що шум є стаціонарним, а знизу наведено результати роботи тесту ADF.




```
    Augmented Dickey-Fuller Test

data:  wn
Dickey-Fuller = -4.8309, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Рисунок 2.10 – Приклад виконання ADF на стаціонарному ряді

Якщо аналіз виконується на стаціонарному ряді, він не потребує додаткових математичних обчислень, як, наприклад, диференціювання, то в якості параметру d у моделі ARIMA(p,d,q) можна вказати 0. Так само 0 можна вказати в якості будь-якого з трьох параметрів, що позначить що цей елемент не повинен використовуватись при побудові моделі. Це дає змогу «перебудувати» ARIMA в простішу модель ARMA, та навіть розкласти на AR, I та MA [40].

Для того, щоб оцінити «якість» підібраних коефіцієнтів моделі ARIMA можна застосувати інформаційний крітерій Акаіке (Akaike information criteria; AIC).

$$AIC = -2 \log(L) + 2(p+q+k+1),$$

де k – кількість параметрів моделі, $\log(L)$ – максимізоване значення функції правдоподібності. Найкращим результатом є модель з найменшим значенням AIC.

Значення AIC буде зростати, якщо буде зростати кількість параметрів k, але буде зменшуватись, якщо негативне максимізоване значення функції зростає. Такий принцип роботи, по суті, і «штрафує» параметри які були надмірно апроксимовані.

Окрім критерію Акаіке також існує інформаційний критерій Баєса (Bayesian information criterion, BIC).

$$BIC = -2 \log(L) + k \log(n),$$

де k – кількість параметрів моделі, $\log(L)$ – максимізоване значення функції правдоподібності, n – кількість спостережень. Принцип роботи AIC та BIC схожий, але BIC має більші «штрафи» за надмірну апроксимацію.

2.3 Сезонна інтегрована модель авторегресії SARIMA

Модель ARIMA підходить у випадках, коли потрібно проаналізувати та зробити прогноз на нестаціонарному часовому ряді з трендом, але вона не підходить для роботи з часовими рядами з сезонністю. Для цього існує розширення моделі яке зветься сезонна інтегрована модель авторегресії (seasonal autoregressive integrated moving average, SARIMA) або сезонна ARIMA [33].

Модель SARIMA формується з параметрів ARIMA, але включаючи додаткові параметри сезонності – SARIMA(p,d,q)(P,D,Q)m [18]. На рисунку 2.11 продемонстровано яка частина параметрів відповідає за сезонні параметри моделі.

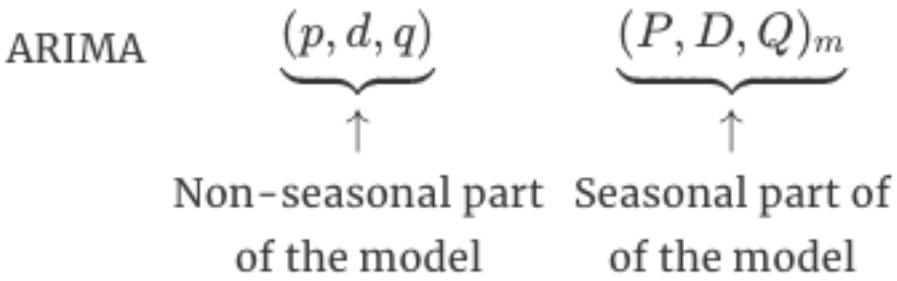


Рисунок 2.11 – Позначення сезонних та несезонних параметрів моделі SARIMA

Для запису формули моделі SARIMA зручніше використовувати форму запису з лаговим оператором [32]. Лаговий оператор дозволяє компактно записати процес диференціації, наприклад

$$y'_t = y_t - y_{t-1} =$$
$$= y_t - L(y_t) =$$
$$= (1 - L)y_t ,$$

де $L(y_t)$ – лаговий оператор. Сезонна різниця між обраним моментом у часі та моментом з попереднього року (враховуючи що в нас річна сезонність) буде мати рівняння

$$L^{12}y_t = y_{t-12} .$$

Загальна форма рівняння моделі SARIMA(p,d,q)(P,D,Q)m:

$$\phi_P(L^m)\varphi(L)\nabla_m^D\nabla^d y_t = \Theta_Q(L^m)\theta(L)\varepsilon_t ,$$

де параметри авторегресії та ковзного середнього представлені поліномами $\varphi(L)$ та $\theta(L)$ порядку p та q, а сезонні параметри авторегресії та ковзного середнього $\phi_P(L^m)$ та $\Theta_Q(L^m)$ порядку P та Q. ∇_m^D та ∇^d – параметри диференціювання звичайних та сезонних даних. L – лаговий оператор, m – сезонність.

$$\phi_P(L^m) = 1 - \phi_1 L^m - \phi_2 L^{2m} - ... - \phi_p L^{pm} ;$$
$$\varphi(L) = 1 - \varphi_1(L) - \varphi_2(L^2) - ... - \varphi_p(L^p);$$
$$\Theta_Q(L^m) = 1 + \Theta_1(L^m) + \Theta_2(L^{2m}) + ... + \Theta_Q(L^{Qm});$$
$$\nabla_m^D = (1 - L^m)^D;$$
$$\nabla^d = (1 - L)^d;$$
$$L^k y_t = y_{t-k}.$$

Підбір параметрів та перевірка на стаціонарність для SARIMA також можна виконувати аналізуючи ACF та PACF графіки, критерію AIC та тесту Бокса-Дженкінса відповідно [38].

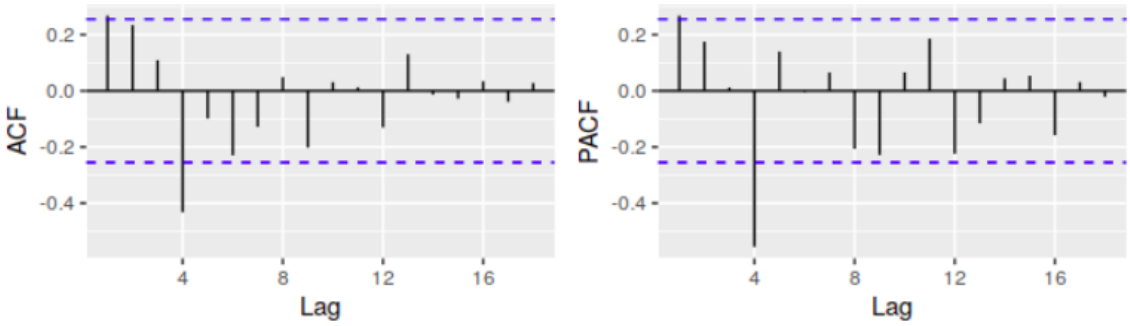


Рисунок 2.12 – Приклад визначення сезонних та несезонних параметрів

Якщо початкові дані мають сезонність, потрібно провести диференціацію за сезоном. Для отриманого результату побудуємо ACF та PACF. Проаналізувавши графіки ACF та PACF з рисунку 2.12 можна побачити що на ACF значення лагу на 1 має «сплеск», що свідчить про те, що це можна враховувати як MA(1) для несезонної частини моделі. Значний «сплеск» на лагу 4 вказує нам, що сезонність цієї моделі – 4. Сезонний параметр MA також буде 1. З цих графіків однаково можна обрати модель SARIMA$(0,1,1)(0,1,1)_4$ керуючись графіком ACF, або модель SARIMA$(1,1,0)(1,1,0)_4$ відповідно до PACF.

Також для порівняння було обрано метод Холта–Вінтерса (метод потрійного експоненційного згладжування), який описано у пункті 1.2.3.3 [9]. Методи сімейства ARIMA та Холта–Вінтерса відрізняються різним підходом до аналізу даних. У моделях ARIMA враховуються параметри автокореляції та комбінуються минулі дані часового ряду та помилки [45]. На відміну від методу Холта–Вінтерса, який фактично є зміненим методу ковзного середнього, тому як у стандартному методу ковзного середнього середнє значення розраховується з заздалегідь визначених минулих спостережень, а у методі Холта–Вінтерса «вага» спостереженням надається не рівномірно – найбільша «вага» присвоюється останнім спостереженням, через те, що вони мають більший вплив на поточні та майбутні спостереження. Через те, що методи ARIMA мають автокореляцію, що надає багато інформації о даних, це є одним із недоліків моделі, тому що це потребує додаткового аналізу даних та додаткової обчислювальної потужності. Тому метод потрійного експоненційного згладжування вважається простіше з точки зору програмної реалізації, тому що він потребує менше інформації та швидше її опрацьовує.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках магістерської роботи було розроблено комп'ютерну програму для прогнозування часових рядів різними моделями. Ця програма призначена для роботи з будь-якою операційною системою. Для реалізації цієї комп'ютерної програми було використано мову програмування Python та середовище розробки PyCharm. Цей вибір обумовлено тим, що PyCharm було створено для роботи з Python, також є підтримка різноманітних фреймворків, до яких входить Django, але у рамках розробки даної програми більш важливою є підтримка Anaconda – дистрибутив Python для аналізу даних.

Серед переваг слід відмітити:

– дебагер;

– комфортну навігацію по коду та файлам;

– auto-complete, який дозволяє запобігти помилкам при зверненні до методів чи класів;

– спеціальний набір засобів, який дозволяє глибше аналізувати та візуалізувати дані (рис. 3.1);

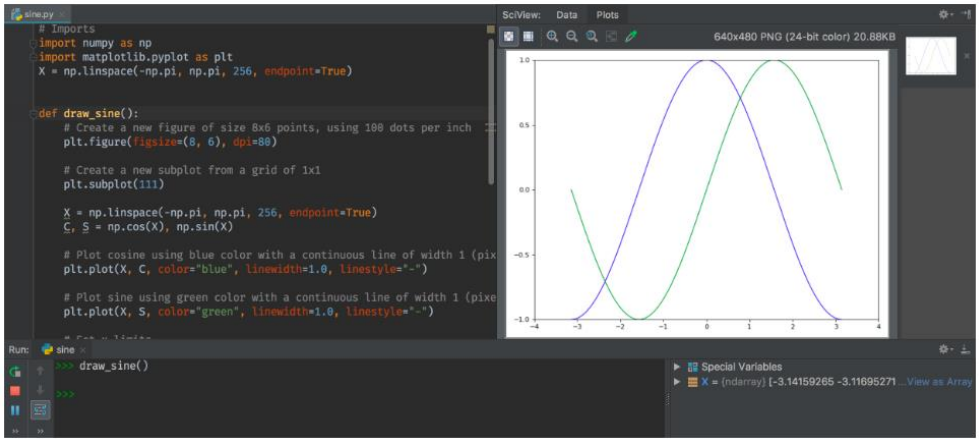


Рисунок 3.1 – Scientific Tools для аналізу та візуалізації даних

– вбудована підтримка наукових бібліотек для аналізу та роботи з даними (рис. 3.2);

– спеціальний графічний інтерфейс для перегляду dataframe або інших типів даних з консолі або графічного дебагеру (рис. 3.3).

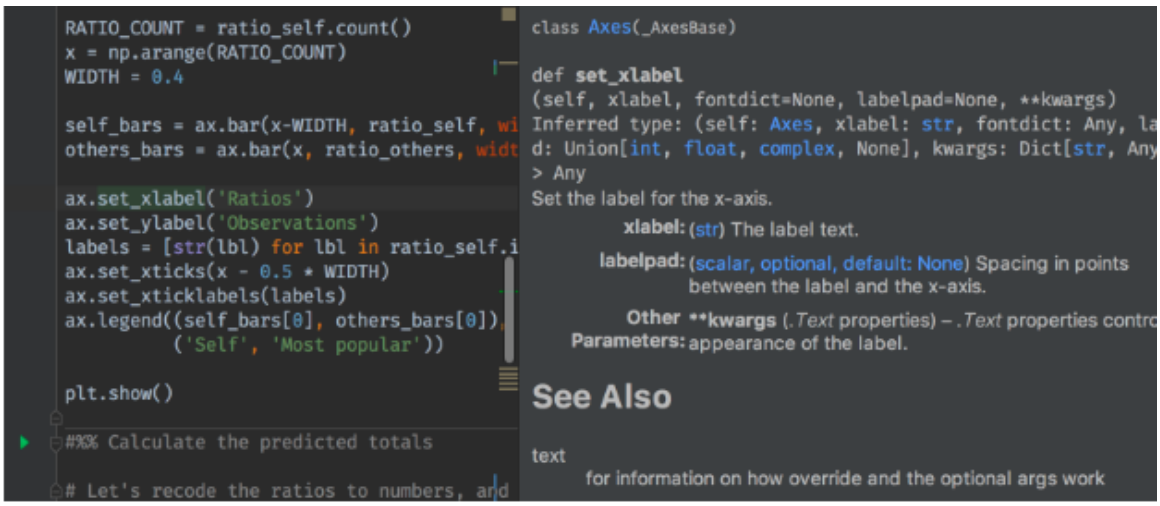


Рисунок 3.2 – Scientific stack support для підтримки наукових бібліотек роботи з даними

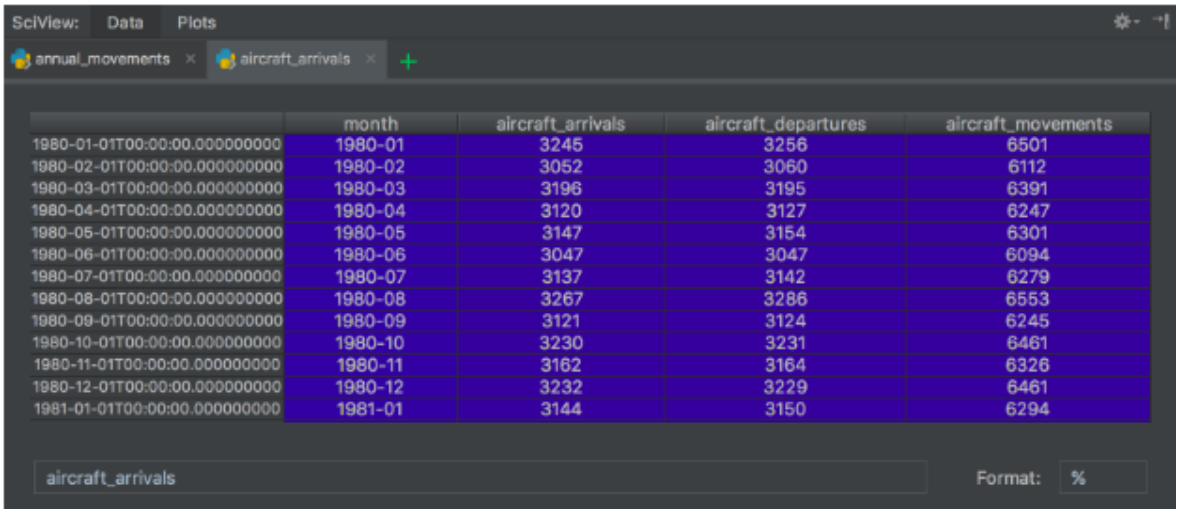


Рисунок 3.3 – SciView для перегляду різних структур даних у комфортному інтерфейсу

3.2 Мова програмування для розробки застосунку

Python – це мова програмування загального призначення та високого рівня, яка може використовуватись для розробки програм на основі графічного інтерфейсу для персональних комп’ютерів, веб сайтів та веб-застосунків. Оскільки це мова програмування високого рівня, вона підтримується основними платформами та системами, що дозволяє запускати код на декількох платформах без необхідності перекомпіляції. Окрім підтримки багатьох платформ, Python має велику стандартну бібліотеку, яка дозволяє використовувати різних спектр модулів, яка дозволяє розробляти функціонал застосунків без написання великої кількості коду, що значно спрощує його читабельність та прискорює процес розробки.

Python є проектом з відкритим та доступним для всіх кодом, крім зниження затрат на розробку та вдосконалення, це також надає можливість для розробників приймати участь у розробці самої мови та створення бібліотек або фреймворків на основи Python без перешкод. Саме через велику кількість бібліотек, що були розроблені під Python та мають гарну інтеграцію зі стандартними можливостями мови програмування, розробники надають перевагу саме Python.

Для написання комп'ютерної програми окрім мови Python було використано:

– numpy – бібліотека яка підтримує роботу з масивами, матрицями та високорівневі математичні функції, які призначені для роботи з багатовимірними масивами ;

– pandas – бібліотека для аналізу та роботи з даними, зокрема використовується для роботи з числовими таблицями та часовими рядами;

– statsmodels – пакет який містить доповнення для бібліотеки scipy для статистичних обчислень, включаючи описову статистику та оцінку та висновки для статистичних моделей;

– scipy – бібліотека призначена для інженерних та научних обчислень (пошук мінімумів та максимумів функцій, обчислення інтегралів);

– matpolib – бібліотека для побудови 2D графіків;

– sklearn – бібліотека яка містить багато алгоритмів для навчання з вчителем та без. Побудована на основі numpy, pandas та matpolib. Включає методи регресіїї, класифікації;

– PySimpleGUI – бібліотека для графічного інтерфейсу;

– simple_salesforce – REST API клієнт для роботи з CRM системами

3.3 Програмна реалізація

У даному застосунку є декілька класів, кожний з яких відповідає за реалізацію відповідного методу. Також є клас з інтерфейсом, який надає користувачу можливість залогінитись до salesforce системи для аналізу даних застосунку. При запуску застосунку користувачу надається можливість ввести свій логін та пароль та додатковий security-token, який гарантує підключення до salesforce.com для отримання даних. Перевірка та підключення здійснюється за допомогою REST API (рис. 3.4, рис. 3.5).

```
def _call_salesforce(self, method, url, **kwargs):
    """Utility method for performing HTTP call to Salesforce.

    Returns a `requests.result` object.
    """
    headers = {
        'Content-Type': 'application/json',
        'Authorization': 'Bearer ' + self.session_id,
        'X-PrettyPrint': '1'
    }
    additional_headers = kwargs.pop('headers', dict())
    headers.update(additional_headers or dict())
    result = self.session.request(method, url, headers=headers, **kwargs)

    if result.status_code >= 300:
        exception_handler(result, self.name)

    sforce_limit_info = result.headers.get('Sforce-Limit-Info')
    if sforce_limit_info:
        self.api_usage = Salesforce.parse_api_usage(sforce_limit_info)

    return result
```

Рисунок 3.4 – Виконання запиту до salesforce.com

```
        self.sf_version = version
        self.domain = domain
        self.session = session or requests.Session()
        self.proxies = self.session.proxies
        # override custom session proxies dance
        if proxies is not None:
            if not session:
                self.session.proxies = self.proxies = proxies
            else:
                logger.warning(
                    'Proxies must be defined on custom session object, '
                    'ignoring proxies: %s', proxies
                )

        # Determine if the user wants to use our username/password auth or pass
        # in their own information
        if all(arg is not None for arg in (
                username, password, security_token)):
            self.auth_type = "password"

            # Pass along the username/password to our login helper
            self.session_id, self.sf_instance = SalesforceLogin(
                session=self.session,
                username=username,
                password=password,
                security_token=security_token,
                sf_version=self.sf_version,
                proxies=self.proxies,
                client_id=client_id,
                domain=self.domain)

        elif all(arg is not None for arg in (
                session_id, instance or instance_url)):
            self.auth_type = "direct"
            self.session_id = session_id

            # If the user provides the full url (as returned by the OAuth
            # interface for example) extract the hostname (which we rely on)
            if instance_url is not None:
                self.sf_instance = urlparse(instance_url).hostname
            else:
                self.sf_instance = instance
```

Рисунок 3.5 – REST API для підключення до salesforce

Після того, як ідентифікація користувача у системе salesforce відбулась успешно програма надає користувачеві доступ до об'єктів бази даних, які зберігають в собі інформацію, щоб користувач здійснив вибір об'єкта який цікавить його. Отримання об'єктів (таблиць) з бази даних здійснюється шляхом відправки SOQL запиту в salesforce систему, який повертає текстовий список всіх таблиць, до яких має доступ користувач (рис. 3.6).

```
    def describe(self):
        """Describes all available objects
        """
        url = self.base_url + "sobjects"
        result = self._call_salesforce('GET', url, name='describe')

        json_result = result.json(object_pairs_hook=OrderedDict)
        if len(json_result) == 0:
            return None

        return json_result
```

```
key_prefix_map = {}
for sobj in sf.describe()['sobjects']:
    if sobj['keyPrefix'] is not None:
        key_prefix_map[sobj['keyPrefix']] = sobj['name']


sf_df = pd.DataFrame(key_prefix_map.values())
```

Рисунок 3.6 – Отримання об'єктів бази даних доступних для користувача

Після отримання результату SOQL запиту – інформації про об'єкти, доступних користувачеві, програма відображає імена всіх об'єктів в текстовому вигляді, щоб користувач міг здійснити вибір таблиці з якою він бажає продовжити роботу (рис. 3.7). Після вибору таблиці відправляється ще один SOQL запит в salesforce для отримання всіх полів обраної таблиці (рис. 3.8).

```
def getObjectFields(obj):
    strObj = obj[0]
    fields = getattr(sf, strObj).describe()['fields']
    flist = [i['name'] for i in fields]
    return flist
```

Рисунок 3.7 – Формування запиту на отримання полів об'єкта на основі вибору користувача

```
    def describe(self, headers=None):
        """Returns the result of a GET to `.../{object_name}/describe` as a
        dict decoded from the JSON payload returned by Salesforce.

        Arguments:

        * headers -- a dict with additional request headers.
        """
        result = self._call_salesforce(
            method='GET', url=urljoin(self.base_url, 'describe'),
            headers=headers
        )
        return result.json(object_pairs_hook=OrderedDict)
```

Рисунок 3.8 – Запит до salesforce на отримання полів

Повернений результат SOQL запиту містить у собі назви всіх полів в текстовому вигляді, які відображаються для користувача з метою вибору полів, які будуть передані в функцію для обробки і подальшого формування прогнозу. На рисунках 3.9 і 3.10 наведені приклади коду, які відповідають за отримання всіх записів з таблиці за обраними користувачем полями і формування з цих даних dataframe, який в подальшому буде оброблений з метою складання моделі прогнозування і графіка прогнозу.

```
def prediction(myList = [], *args):
    combineQuery = 'SELECT '

    for i, val in enumerate(myList):
        print(myList[i])
        combineQuery += myList[i]
        if i < len(myList)-1:
            combineQuery += ','


    sf_select = sf.query_all((combineQuery + ' FROM ' + globalobjectName))
    sf_df = pd.DataFrame(sf_select['records']).drop(columns='attributes')

    for i, name in enumerate(list(sf_df)):
        if name.endswith('__c'):
            print(name[:len(name)-3])
            sf_df = sf_df.rename(columns={name: name[:len(name)-3]})
```

Рисунок 3.9 – Формування запиту на основі обраної користувачем таблиці

```
    def query_all(self, query, include_deleted=False, **kwargs):
        """
        Arguments

        * query -- the SOQL query to send to Salesforce, e.g.
                   SELECT Id FROM Lead WHERE Email = "waldo@somewhere.com"
        * include_deleted -- True if the query should include deleted records.
        """

        result = self.query(query, include_deleted=include_deleted, **kwargs)
        all_records = []

        while True:
            all_records.extend(result['records'])
            # fetch next batch if we're not done else break out of loop
            if not result['done']:
                result = self.query_more(result['nextRecordsUrl'],
                                         identifier_is_url=True)
            else:
                break

        result['records'] = all_records
        return result
```

Рисунок 3.10– Запит до salesforce на отримання записів

Для прогнозування використовуються методи ARMA, ARIMA, SARIMA і Холта-Вінтерса.

Для моделі ARMA дані, після формування dataframe, передаються в функцію для декомпозиції і аналізу отриманого графіка, також виконується тест Діккі Фуллера, для перевірки їх стаціонарності (рис. 3.11, рис. 3.12).

```
result = seasonal_decompose(sf_df, model='multiplicative')
fig = result.plot()
plt.show()
```

Рисунок 3.11 – Декомпозиція даних

```
result = adfuller(training_data['Quantity'])
print('ADF Statistic: {}'.format(result[0]))
print('p-value: {}'.format(result[1]))
print('Critical Values:')
for key, value in result[4].items():
    print('\t{}: {}'.format(key, value))
```

```
    xdall = lagmat(xdiff[:, None], bestlag, trim='both', original='in')
    nobs = xdall.shape[0]
    xdall[:, 0] = x[-nobs - 1:-1]  # replace 0 xdiff with level of x
    xdshort = xdiff[-nobs:]
    usedlag = bestlag
else:
    usedlag = maxlag
    icbest = None
if regression != 'nc':
    resols = OLS(xdshort, add_trend(xdall[:, :usedlag + 1],
                 regression)).fit()
else:
    resols = OLS(xdshort, xdall[:, :usedlag + 1]).fit()

adfstat = resols.tvalues[0]


# Get approx p-value and critical values
pvalue = mackinnonp(adfstat, regression=regression, N=1)
critvalues = mackinnoncrit(N=1, regression=regression, nobs=nobs)
critvalues = {"1%" : critvalues[0], "5%" : critvalues[1],
              "10%" : critvalues[2]}
if store:
    resstore.resols = resols
    resstore.maxlag = maxlag
    resstore.usedlag = usedlag
    resstore.adfstat = adfstat
    resstore.critvalues = critvalues
    resstore.nobs = nobs
    resstore.H0 = ("The coefficient on the lagged level equals 1 - "
                   "unit root")
    resstore.HA = "The coefficient on the lagged level < 1 - stationary"
    resstore.icbest = icbest
    resstore._str = 'Augmented Dickey-Fuller Test Results'
    return adfstat, pvalue, critvalues, resstore
else:
    if not autolag:
        return adfstat, pvalue, usedlag, nobs, critvalues
```

Рисунок 3.12 – Тест Діккі-Фуллера

Після проведення тесту, в разі, якщо часовий ряд нестаціонарний, проводяться операції логарифмування і диференціації (рис. 3.14), для приведення ряду до стаціонарності. Якщо ж після тесту результат свідчить про те, що ряд стаціонарний, логаріфмувати та диференціювати його не потрібно. Далі будуються графіки автокорреляції і часткової автокореляції для визначення параметрів AR і MA (рис. 3.13).

```
acfDIFF = plot_acf(training_data, lags=20, title='ACF', alpha=0.5) #1
acfDIFF.show()
pacfDIFF = plot_pacf(training_data, lags=20, title='PACF', alpha=0.5) #1
pacfDIFF.show()
```

Рисунок 3.13 – Функції автокореляції та часткової автокореляції

```
print('--------LOG----------')

to_log_data = np.log(sf_df['Sales'])
result = adfuller(to_log_data)
print('ADF Statistic: {}'.format(result[0]))
print('p-value: {}'.format(result[1]))
print('Critical Values:')
for key, value in result[4].items():
    print('\t{}: {}'.format(key, value))


print('----------------------')

print('--------DIFF----------')

diff_data = np.diff(sf_df['Sales'])
result = adfuller(diff_data)
print('ADF Statistic: {}'.format(result[0]))
print('p-value: {}'.format(result[1]))
print('Critical Values:')
for key, value in result[4].items():
    print('\t{}: {}'.format(key, value))


print('----------------------')

print('--------LOG/DIFF----------')

log_diff = np.log(sf_df['Sales']).diff().dropna()
result = adfuller(log_diff)
print('ADF Statistic: {}'.format(result[0]))
print('p-value: {}'.format(result[1]))
print('Critical Values:')
for key, value in result[4].items():
    print('\t{}: {}'.format(key, value))


print('----------------------')
```

Рисунок 3.14 – Логарифмування і диференціація

Крім візуального оцінювання графіків автокорреляции і часткової автокореляції можливо використовувати вбудовану функцію, яка здійснює підбір параметрів і видає найбільш підходящий до моделі (рис. 3.15).

```
res = sm.tsa.arma_order_select_ic(diff_data, ic=['aic', 'bic'], trend='nc')
print(res.aic_min_order)
print(res.bic_min_order)
```

Рисунок 3.15 – AIC BIC для ARMA

На основі підібраних параметрів можна будувати модель (рис. 3.16).

```
model = sm.tsa.ARMA(carSalesData, (4, 1)).fit()
pred = model.plot_predict(start='2014-01-01', end='2017-01-01', dynamic=False)
pred.show()
```

Рисунок 3.16 – Підбір моделі і побудова графіку прогнозу

Оцінити точність прогнозу можна за допомогою графіку, на якому відображається крива прогнозу і реальні значення, а також можна провести оцінку за допомогою алгоритмів помилок MSE і RMSE (рис. 3.17).

```
y_forecasted = model.predict(start = '2014-01-01', end = '2017-01-01', dynamic=False)
print(y_forecasted)
print(carSalesData['2014-01-01': '2017-01-01'])
actual_data = carSalesData['2014-01-01': '2017-01-01']
y_truth = actual_data
mse = mean_squared_error(y_truth, y_forecasted)
print(mse)
print(np.sqrt(mse))
```

Рисунок 3.17 – Розрахунок MSE і RMSE для прогнозу

Для моделей ARIMA і SARIMA приведення до стаціонарності не є обов'язковим, так як ці моделі передбачають перевірку параметрів сезонності і диференціацію, як показано на рисунку 3.18 для SARIMA. Для моделі ARIMA використовується функція auto_arima, замість SARIMAX.

```
AIC_list = pd.DataFrame({}, columns=['param','param_seasonal','AIC'])
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(training_data,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)

            results = mod.fit()

            print('ARIMA{}x{} - AIC:{}'.format(param, param_seasonal, results.aic))
            temp = pd.DataFrame([[param,  param_seasonal, results.aic]], columns=['param','param_seasonal','AIC'])
            AIC_list = AIC_list.append(temp, ignore_index=True)
            del temp
        except:
            continue

m = np.nanmin(AIC_list['AIC'].values)# Find minimum value in AIC
l = AIC_list['AIC'].tolist().index(m)# Find index number for lowest AIC
Min_AIC_list = AIC_list.iloc[l,:]
```

Рисунок 3.18 – Функція підбору параметрів моделі SARIMA

Для методу потрійного експоненційного згладжування або ж методу Холта-Вінтерса так само необхідно підібрати параметри, тільки в даному випадку це параметри

тренда, сезонності і рівня (рис. 3.19). Наводити ряд до стаціонарності не є необхідною умовою

```
    alpha = float(smoothing_level) if smoothing_level is not None else None
    beta = float(smoothing_slope) if smoothing_slope is not None else None
    gamma = float(smoothing_seasonal) if smoothing_seasonal is not None else None
    phi = float(damping_slope) if damping_slope is not None else None
    self._l0 = float(initial_level) if initial_level is not None else None
    self._b0 = float(initial_slope) if initial_slope is not None else None

    data = self.endog
    damped = self.damped
    seasoning = self.seasoning
    trending = self.trending
    trend = self.trend
    seasonal = self.seasonal
    m = self.seasonal_periods
    opt = None
    phi = phi if damped else 1.0
    if use_boxcox == 'log':
        lamda = 0.0
        y = boxcox(data, lamda)
    elif isinstance(use_boxcox, float):
        lamda = use_boxcox
        y = boxcox(data, lamda)
    elif use_boxcox:
        y, lamda = boxcox(data)
    else:
        lamda = None
        y = data.squeeze()
    if np.ndim(y) != 1:
        raise ValueError('Only 1 dimensional data supported')
    self._y = y = np.ascontiguousarray(y, dtype=np.double)
    lvls = np.zeros(self.nobs)
    b = np.zeros(self.nobs)
    s = np.zeros(self.nobs + m - 1)
    p = np.zeros(6 + m)
    max_seen = np.finfo(np.double).max
    l0, b0, s0 = self.initial_values()

    xi = np.zeros_like(p, dtype=np.bool)
    if optimized:
        init_alpha = alpha if alpha is not None else 0.5 / max(m, 1)
        init_beta = beta if beta is not None else 0.1 * init_alpha if trending else beta
        init_gamma = None
        init_phi = phi if phi is not None else 0.99
```

Рисунок 3.19 – Підбір параметрів тренда, сезонності і рівня для набору даних

Модель з підібраних параметрів можна оцінити за допомогою SSE, чим менше значення SSE тим краще підібрані параметри. Параметри які дали найменшу помилку вибираються для побудови моделі прогнозування (рис. 3.20, рис. 3.21). Оцінка підібраною моделі здійснюється при використанні AIC.

```
model = ExponentialSmoothing(training_data, seasonal='mul', seasonal_periods=12).fit(use_brute = True)
pred = model.predict(start='2014-01-01', end='2017-01-01')
```

Рисунок 3.20 – Встановлення часового проміжку для прогнозування

```
aicc = aic + aicc_penalty
bic = self.nobs * np.log(sse / self.nobs) + k * np.log(self.nobs)
resid = data - fitted[:-h - 1]
if remove_bias:
    fitted += resid.mean()
self.params = {'smoothing_level': alpha,
               'smoothing_slope': beta,
               'smoothing_seasonal': gamma,
               'damping_slope': phi if damped else np.nan,
               'initial_level': lvls[0],
               'initial_slope': b[0] / phi,
               'initial_seasons': s[:m],
               'use_boxcox': use_boxcox,
               'lamda': lamda,
               'remove_bias': remove_bias}

# Format parameters into a DataFrame
codes = ['alpha', 'beta', 'gamma', 'l.0', 'b.0', 'phi']
codes += ['s.{0}'.format(i) for i in range(m)]
idx = ['smoothing_level', 'smoothing_slope', 'smoothing_seasonal',
       'initial_level', 'initial_slope', 'damping_slope']
idx += ['initial_seasons.{0}'.format(i) for i in range(m)]

formatted = [alpha, beta, gamma, lvls[0], b[0], phi]
formatted += s[:m].tolist()
formatted = list(map(lambda v: np.nan if v is None else v, formatted))
formatted = np.array(formatted)
if is_optimized is None:
    optimized = np.zeros(len(codes), dtype=np.bool)
else:
    optimized = is_optimized.astype(np.bool)
included = [True, trending, seasoning, True, trending, damped]
included += [True] * m
formatted = pd.DataFrame([[c, f, o] for c, f, o in zip(codes, formatted, optimized)],
                         columns=['name', 'param', 'optimized'],
                         index=idx)
formatted = formatted.loc[included]

hwfit = HoltWintersResults(self, self.params, fittedfcast=fitted,
                           fittedvalues=fitted[:-h - 1], fcastvalues=fitted[-h - 1:],
                           sse=sse, level=level, slope=slope, season=season, aic=aic,
                           bic=bic, aicc=aicc, resid=resid, k=k,
                           params_formatted=formatted, optimized=optimized)
```

Рисунок 3.21 – Прогнозування за допомогою методу Холта–Вінтерса

3.4 Порівняння результатів прогнозування

Для прогнозування були обрані кілька наборів даних:

– щомісячні дані про продаж автомобілів в період з 2007 до 2017 року, які мають 2694 записи;

– щомісячні дані про продаж автомобілів в період з 2016 по 2018 рік, які мають 36 записів;

– щотижневі дані про ціну авокадо в період з 2015 по 2018 рік, які мають 18249 записів.

Для даних про продаж автомобілів в період з 2007 до 2017 року графік показано на рисунку 3.22.

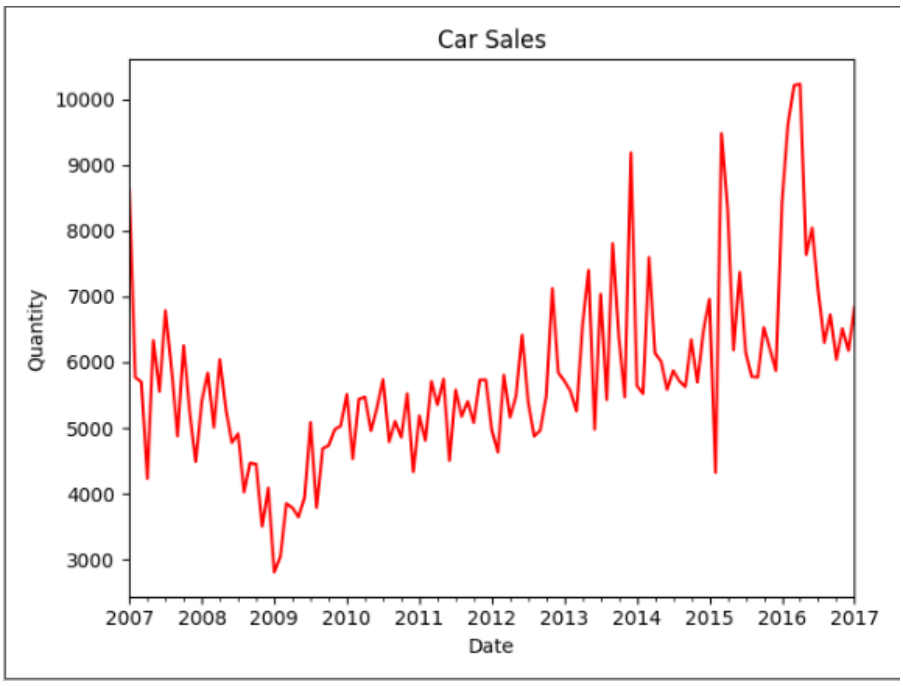


Рисунок 3.22 – Графік вихідних даних

На рисунку 3.22 можна помітити деякі сильні сплески значень продажів під кінець року і то що з роками кількість продажів в загальному збільшується. Це свідчить про наявність тренда і, можливо, сезонності. Для того щоб упевнитися зробимо декомпозицію даних - розкладемо на такі компоненти як тренд, сезонність і залишкові дані.

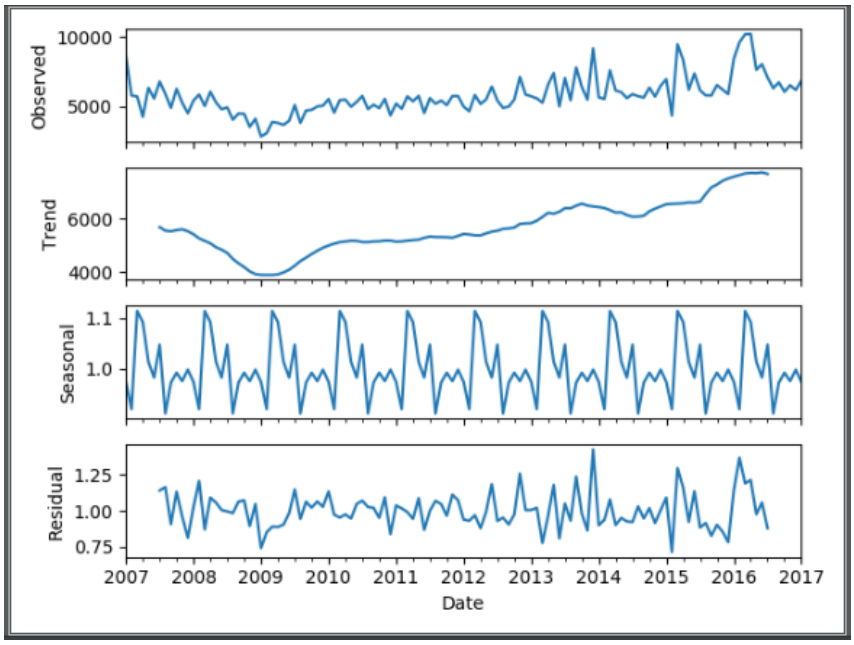


Рисунок 3.23 – Графік декомпозиції даних

На рисунку 3.23 ми бачимо декопозовані дані. З цього графіку можна зробити висновок, що дані мають тренд і мають сезонність. Тобто нам потрібна модель, яка враховує і тренд і сезонність. Для цього підійде SARIMA та модель Холта-Вінтерса.

Для методу SARIMA стационарность даних не є обов'язковою умовою і наявність тренда і сезонності свідчить про те, що дані не стаціонарні, так само це можна спостерігати на графіку вихідних даних.

А на рисунку 3.24 дані представлені в стаціонарному вигляді.

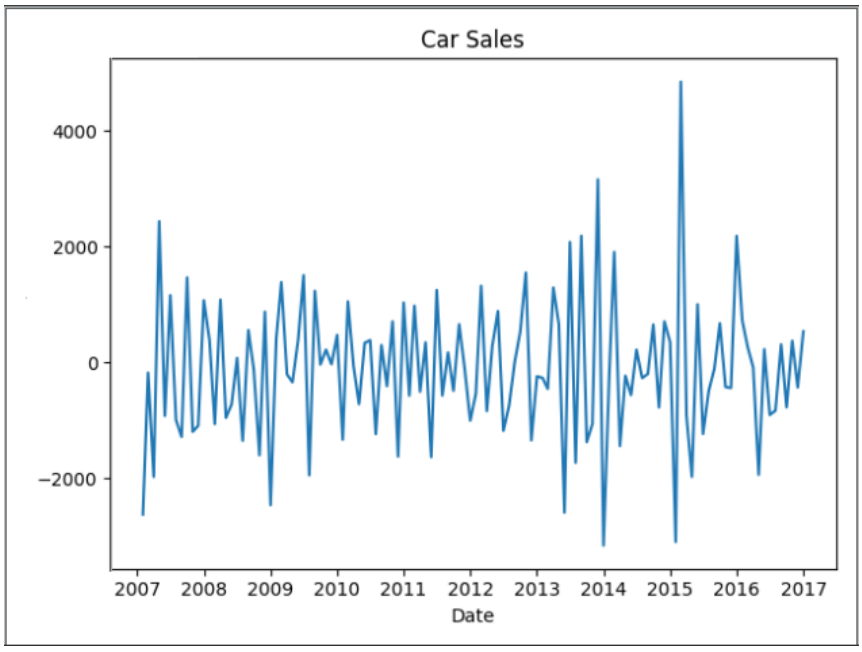


Рисунок 3.24 – Дані в стаціонарному вигляді

Даними для тренування обрані дані про продажі з 2007 по 2016 рік. Прогноз буде виконаний з 2016 по 2017 рік.

В результаті запуску алгоритму SARIMA на даних, алгоритм перевірив 80 комбінацій сезонних і несезонних параметрів, для вибору найменшого значення AIC (рис 3.25).

	param	param_seasonal	AIC
0	(0, 0, 0)	(0, 0, 0, 12)	2174.031540
1	(0, 0, 0)	(0, 1, 0, 12)	1644.961190
2	(0, 0, 0)	(0, 2, 0, 12)	1523.414835
3	(0, 0, 0)	(1, 0, 0, 12)	1669.421087
4	(0, 0, 0)	(1, 1, 0, 12)	1449.976032
5	(0, 0, 0)	(1, 2, 0, 12)	1287.158089
6	(0, 0, 0)	(2, 0, 0, 12)	1449.239066
7	(0, 0, 0)	(2, 1, 0, 12)	1231.993901
8	(0, 0, 0)	(2, 2, 0, 12)	1049.794666
9	(0, 1, 0)	(0, 0, 0, 12)	1823.545628
10	(0, 1, 0)	(0, 1, 0, 12)	1669.814199
11	(0, 1, 0)	(0, 2, 0, 12)	1545.654233
12	(0, 1, 0)	(1, 0, 0, 12)	1638.927810
13	(0, 1, 0)	(1, 1, 0, 12)	1463.622439
14	(0, 1, 0)	(1, 2, 0, 12)	1301.082847
15	(0, 1, 0)	(2, 0, 0, 12)	1443.122317
16	(0, 1, 0)	(2, 1, 0, 12)	1254.177243
17	(0, 1, 0)	(2, 2, 0, 12)	1081.759858
18	(0, 2, 0)	(0, 0, 0, 12)	1924.724938
19	(0, 2, 0)	(0, 1, 0, 12)	1761.753854

Рисунок 3.25 – Приклад підбору параметрів SARIMA

В результаті порівняння AIC всіх пар значень, як несезонних параметрів було вибрано (2,2,0), в якості сезонних (2.2,0) і сезонність 12, так як однаковий сплеск зростання продажів можна помітити щороку взимку, AIC для цих значень 999.582. Отже у нас вийшла модель SARIMA$(2,2,0)(2,2,0)_{12}$. Результати прогнозування можна побачити на рисунку 3.26.

2016-01-01	7072.129884
2016-02-01	4353.863958
2016-03-01	10013.709294
2016-04-01	11176.027404
2016-05-01	9089.441422
2016-06-01	7524.236831
2016-07-01	7663.204082
2016-08-01	7325.149893
2016-09-01	6831.290121
2016-10-01	7452.484623
2016-11-01	5786.740922
2016-12-01	6207.394055
2017-01-01	8978.169191

2016-01-01	8441
2016-02-01	9620
2016-03-01	10211
2016-04-01	10231
2016-05-01	7637
2016-06-01	8045
2016-07-01	7077
2016-08-01	6299
2016-09-01	6724
2016-10-01	6041
2016-11-01	6512
2016-12-01	6178
2017-01-01	6834

Рисунок 3.26 – Справа спрогнозовані значення, зліва реальні значення

Візуально прогноз виглядає як на рисунку 3.27.

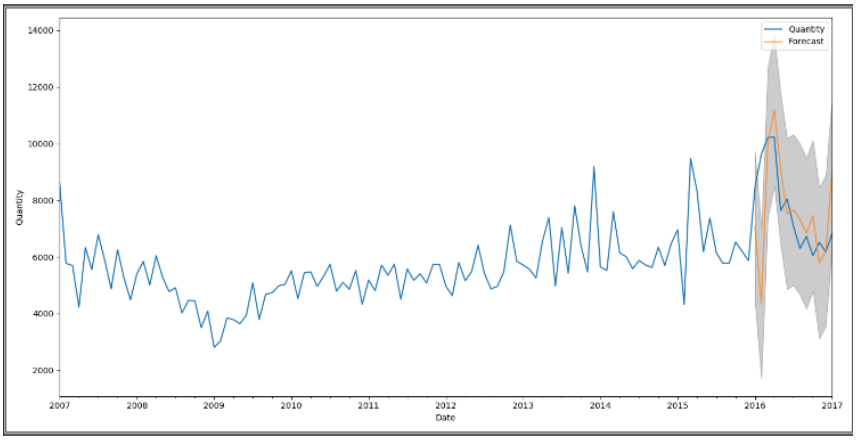


Рисунок 3.27 – Графічний вид прогнозу

Так само ми можемо провести аналіз залишків. Для цього ми можемо побудувати графік Q-Q (quantile-quantile), гістограму нормального розподілу і коррелограму залишків.

Якщо аналіз залишків покаже, що розподіл не є гаусовим (нормальним), то це може свідчити про те, що прогноз обраною моделлю, можливо, неправильний і слід вибрати іншу модель.

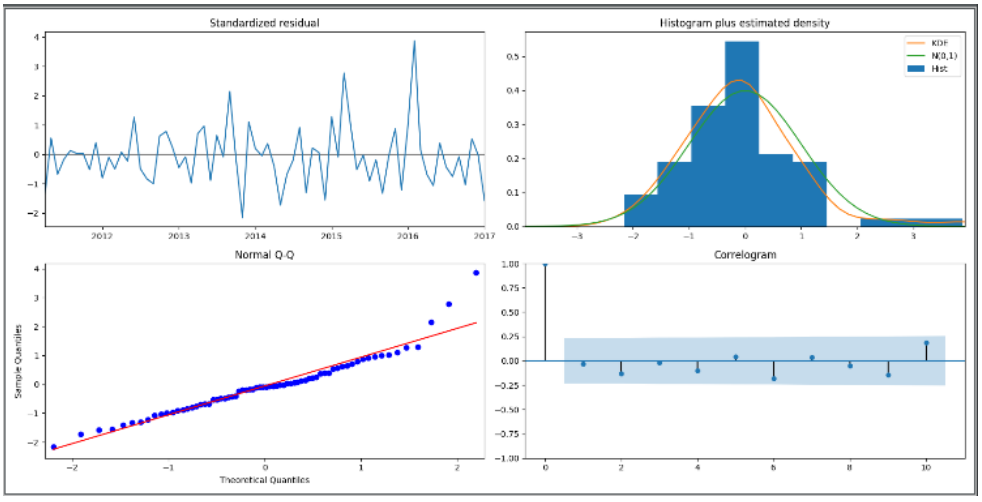


Рисунок 3.28 – Аналіз залишків даних

На рисунку 3.28 видно, що гістограма і щільність розподілу відповідають нормальному розподілу. Графік Q-Q показує, що значення результату прогнозу (точки) лягають на лінію, яка є нормальним розподілом. І гістограма і графік квантилів свідчать про те, що отримані результати відповідають нормальному розподілу. Коррелограмма показує, що аномальних значень які виходять за межі немає. Тому ми можемо вважати, що результати даної моделі прийнятні.

Середній квадрат помилки розподілу (MSE) для SARIMA = 3187963.083907801, середньоквадратичне відхилення (RMSE) = 1785.4867918603602. Час роботи алгоритму – 34.98427605628967 секунди.

Для методу Холта-Вінтерса мінімальне значення AIC - 1690.183, що продемонстровано на рисунку 3.29.

ExponentialSmoothing Model Results

Dep. Variable:	endog	No. Observations:	121
Model:	ExponentialSmoothing	SSE	106457195.249
Optimized:	True	AIC	1690.183
Trend:	Multiplicative	BIC	1737.711
Seasonal:	Multiplicative	AICC	1697.708
Seasonal Periods:	12	Date:	Sun, 17 Nov 2019
Box-Cox:	False	Time:	16:23:59
Box-Cox Coeff.:	None		

Рисунок 3.29 – Значення AIC для Холта-Вінтерса

Оптимальні значення alpha, beta, gamma - 0.3294017, 0.0525675 та 0.4403398 відповідно (рис. 3.30).

	coeff	code	optimized
smoothing_level	0.3294017	alpha	True
smoothing_slope	0.0525675	beta	True
smoothing_seasonal	0.4403398	gamma	True
initial_level	6008.2728	l.0	True
initial_slope	1.0000000	b.0	True
damping_slope	0.000000	phi	True
initial_seasons.0	1.0999707	s.0	True
initial_seasons.1	1.0031479	s.1	True
initial_seasons.2	1.1975766	s.2	True
initial_seasons.3	1.1550351	s.3	True
initial_seasons.4	1.0779111	s.4	True
initial_seasons.5	1.0476326	s.5	True
initial_seasons.6	1.0826272	s.6	True
initial_seasons.7	0.9514358	s.7	True
initial_seasons.8	1.0167389	s.8	True
initial_seasons.9	1.0277853	s.9	True
initial_seasons.10	1.0245639	s.10	True
initial_seasons.11	1.0616927	s.11	True

Рисунок 3.30 – Параметри alpha, beta, gamma

На рисунку 3.31 продемонстровані спрогнозовані і реальні значення за 2016 - 2017 рік.

2016-01-01	6508.014403
2016-02-01	6515.840934
2016-03-01	8999.430227
2016-04-01	9064.659818
2016-05-01	8817.935710
2016-06-01	8192.164691
2016-07-01	8415.715967
2016-08-01	7008.371672
2016-09-01	7239.694736
2016-10-01	7146.634162
2016-11-01	6761.177942
2016-12-01	6921.139237
2017-01-01	6917.055502

2016-01-01	8441
2016-02-01	9620
2016-03-01	10211
2016-04-01	10231
2016-05-01	7637
2016-06-01	8045
2016-07-01	7077
2016-08-01	6299
2016-09-01	6724
2016-10-01	6041
2016-11-01	6512
2016-12-01	6178
2017-01-01	6834

Рисунок 3.31 – Зліва спрогнозовані дані, праворуч реальні дані

Графік отриманого прогнозу продемонстровано на рисунку 3.32.

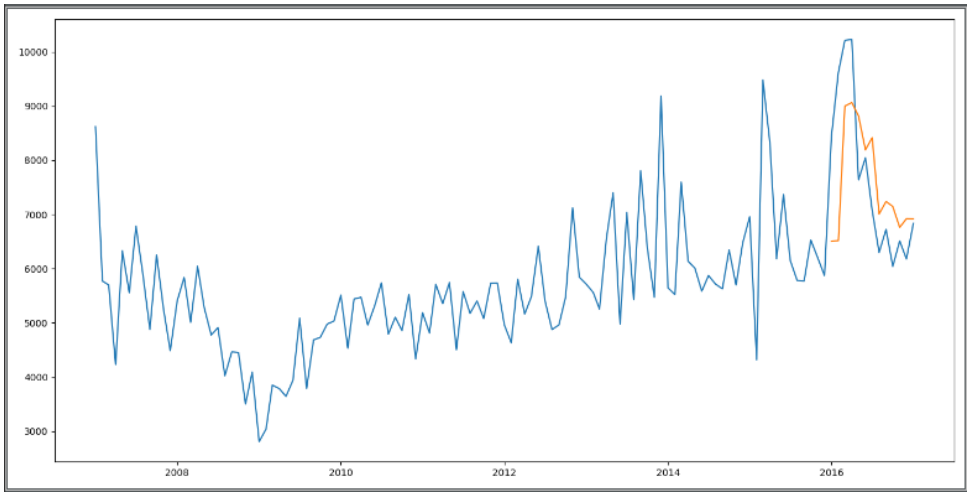


Рисунок 3.32 – Графік прогнозу методом Холта-Вінтерса

Середній квадрат помилки розподілу (MSE) = 1693979.5785449299, середньоквадратичне відхилення (RMSE) = 1301.5297071311627. Час роботи алгоритму – 1.0705358982086182 секунд.

Виходячи з результатів, отриманих цими двома методами, можна стверджувати, що метод Холта-Вінтерса є більш ефективним, так як середньоквадратичне відхилення менше ніж у SARIMA і час роботи набагато швидше.

Так само можна провести порівняння з методами ARMA і ARIMA.

Для ARMA додатково буде потрібно привести ряд до стаціонарного виду за допомогою диференціації і логарифмирования (рис. 3.33).

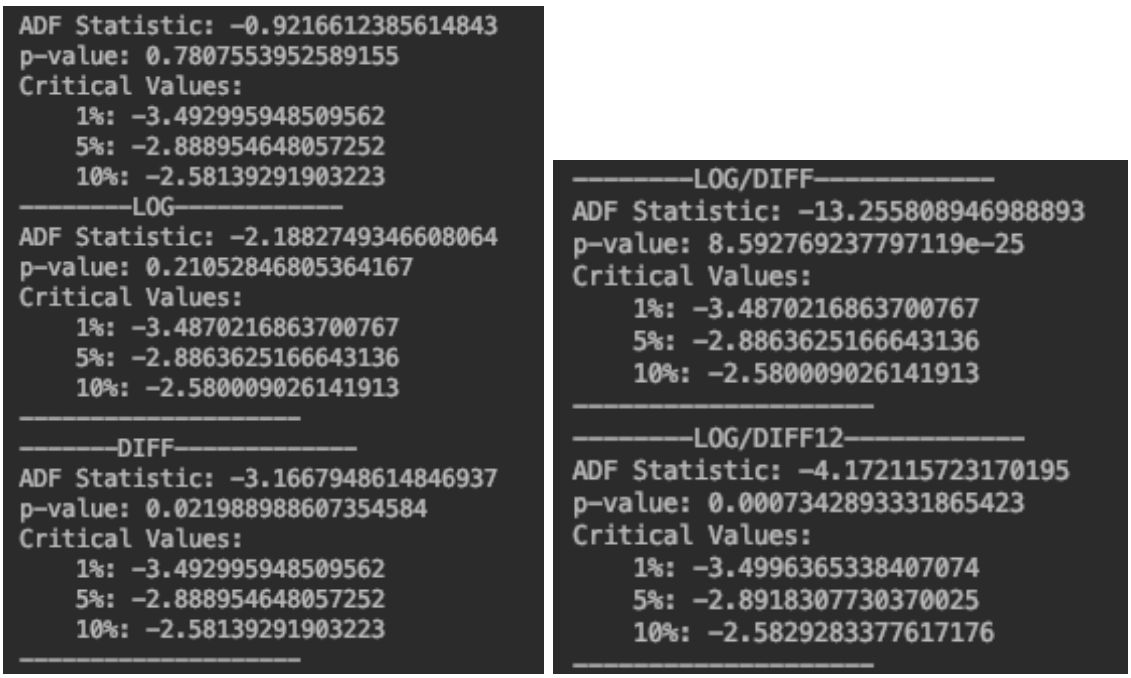


Рисунок 3.33 – Приведення до стаціонарності

Після застосування операції диференціірованія дані стали стаціонарними. Отже можна приступити до підбору параметрів (рис. 3.34).

```
res = sm.tsa.arma_order_select_ic(diff_data, ic=['aic', 'bic'], trend='nc')
print(res.aic_min_order)
```

Рисунок 3.34 – Підбір параметрів

Оптимальними параметрами для ARMA є (0,1). На рисунках 3.35 і 3.36 можна побачити чисельний і графічний результат прогнозу.

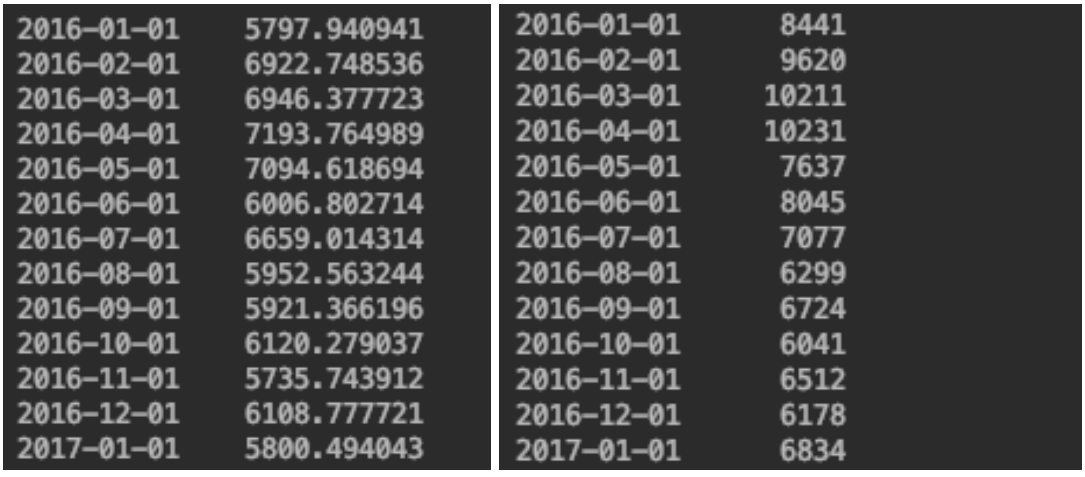


Рисунок 3.35 - Результати прогнозу

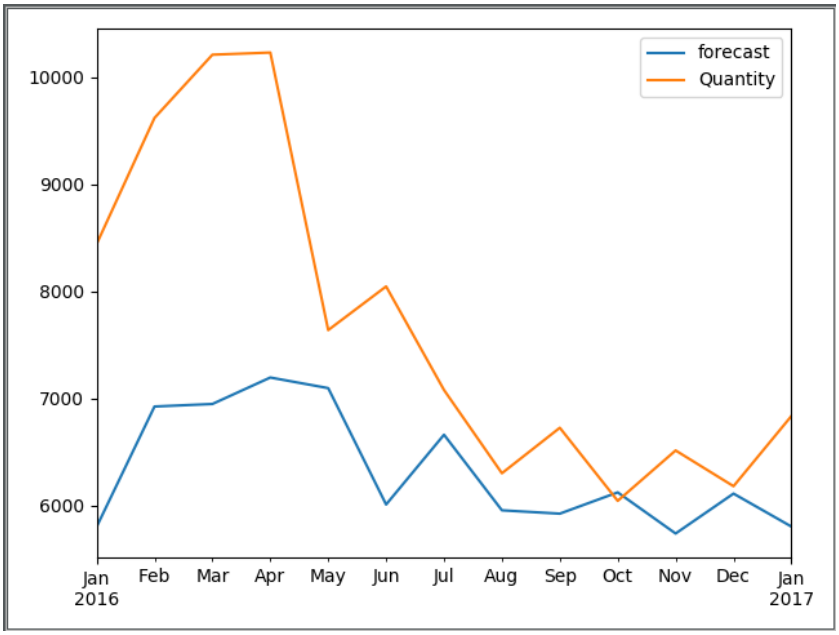


Рисунок 3.36 – Графік прогнозу ARMA

Середній квадрат помилки розподілу (MSE) = 4062076.4028056245, середньоквадратичне відхилення (RMSE) = 2015.4593528041255. Час роботи алгоритму – 5.379272937774658 секунд.

Прогноз вийшов значно гірше за значенням RMSE так як дана модель даних не враховує сезонність, яка впливає на прогнозування. Але час роботи алгоритму менше, ніж у SARIMA, так як приведення до стаціонарності не було проведено програмно і кількість параметрів для підбору менше.

Для ARIMA результати продемонстровані на рисунках 3.37 і 3.38.

2016-01-01	6147.094617
2016-02-01	5099.334856
2016-03-01	7407.404015
2016-04-01	6814.754498
2016-05-01	6270.866649
2016-06-01	6208.282997
2016-07-01	6243.086017
2016-08-01	5638.212830
2016-09-01	6028.098623
2016-10-01	6261.771910
2016-11-01	6080.450661
2016-12-01	6499.539268
2017-01-01	6215.008062

2016-01-01	8441
2016-02-01	9620
2016-03-01	10211
2016-04-01	10231
2016-05-01	7637
2016-06-01	8045
2016-07-01	7077
2016-08-01	6299
2016-09-01	6724
2016-10-01	6041
2016-11-01	6512
2016-12-01	6178
2017-01-01	6834

Рисунок 3.37 – Зліва спрогнозовані значення, праворуч реальні

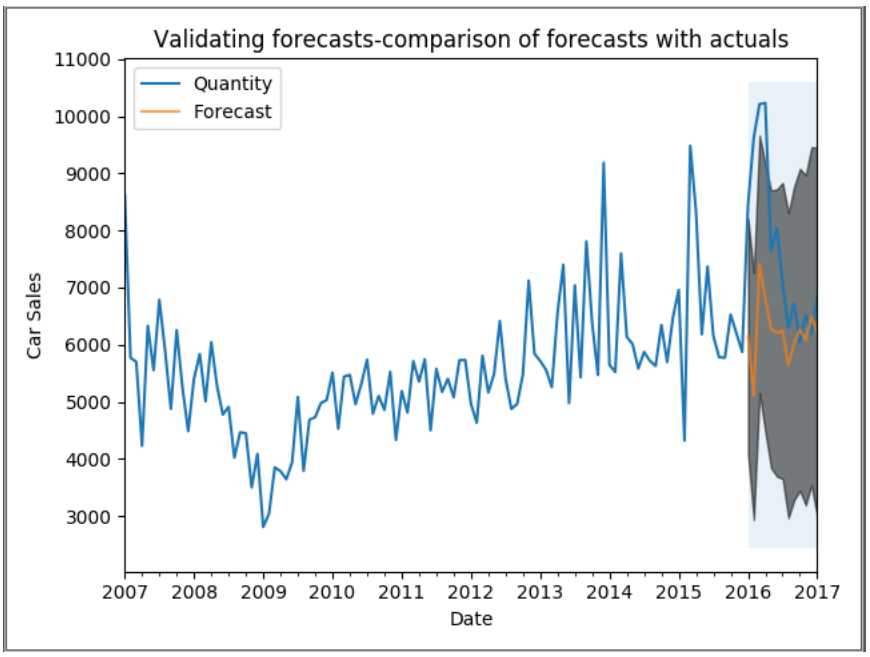


Рисунок 3.38 – Графік прогнозу

Середній квадрат помилки розподілу (MSE) = 3170203.3754457813, середньоквадратичне відхилення (RMSE) = 1780.5064940757113. Час роботи алгоритму – 5.434921026229858 секунд.

ARIMA показала меншу помилку, ніж у ARMA, але більшу ніж у SARIMA і Холта-Вінтерса.

Результати роботи алгоритмів на даних про продаж автомобілів в період з 2016 по 2018 продемонстровані на рисунку 3.40. Тестовими даними були дані в період з 2016 по 2018 рік. Прогноз виконувався з 2017 по 2018 рік.

Декомпозиція даних показана на рисунку 3.39 показує, що ці дані мають тренд і сезонність.

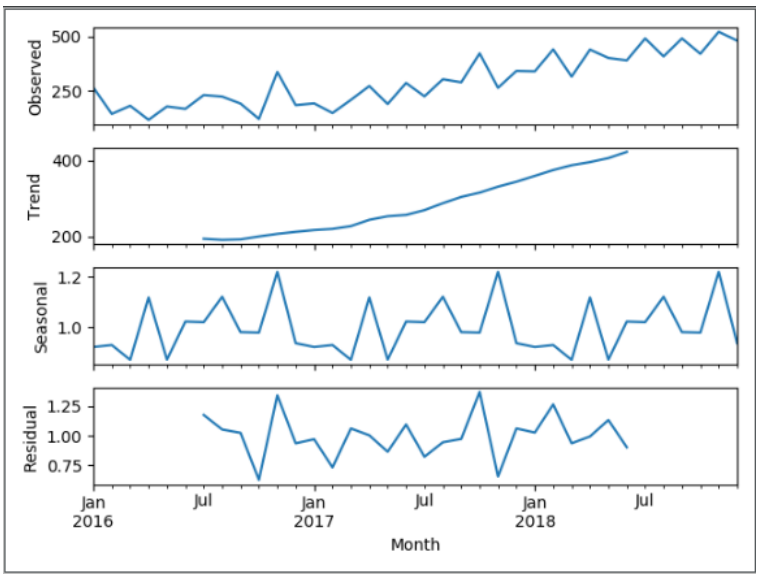


Рисунок 3.39 – Декомпозиція даних

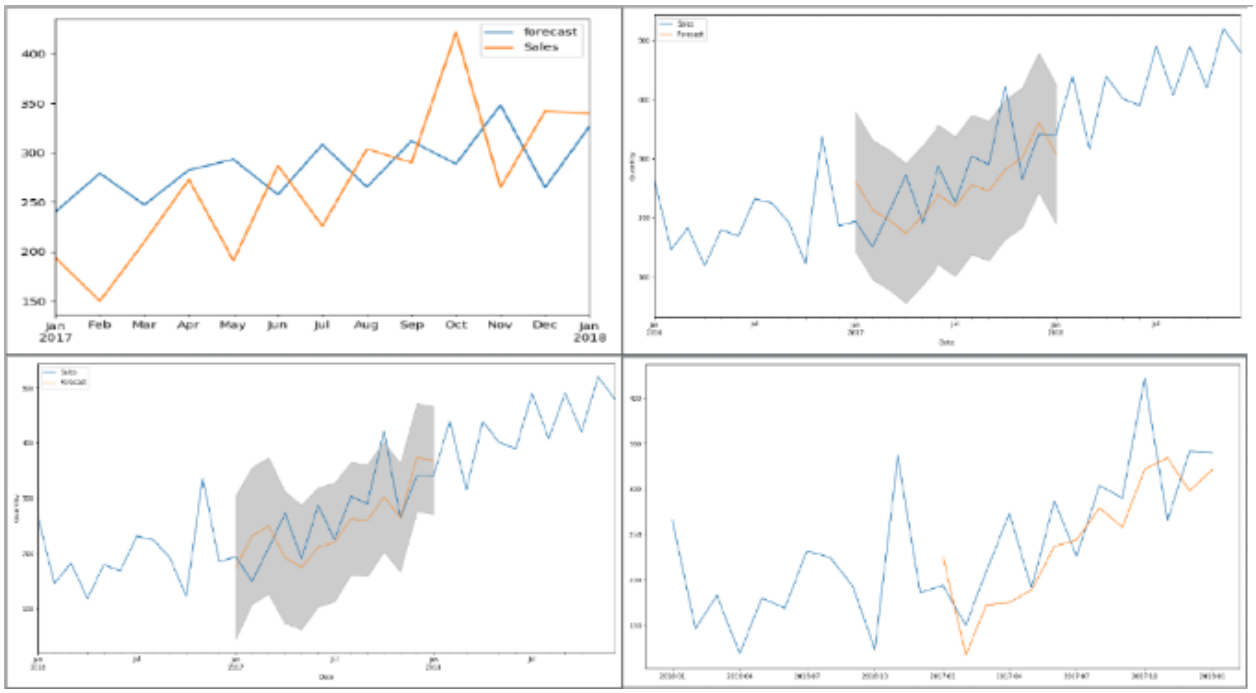


Рисунок 3.40 – Результати прогнозу зліва зверху ARMA, ARIMA, SARIMA, Holt-Winters

Час роботи і значення середнього квадрату помилки розподілу (MSE), середньоквадратичного відхилення (RMSE) і час роботи наведені в таблиці 3.1.

Таблиця 3.1 – Результати помилок та час роботи для моделей прогнозування

Модель	MSE	RMSE	Час роботи, сек
ARMA	5482.969408571299	74.0470756247085	3.946199655532837
ARIMA	3692.1699377032046	60.76322849967079	3.3478260040283203
SARIMA	3059.4659177788835	55.3124390872332	4.662951946258545
Holt-Winters	2666.8783640405545	51.64182765976195	2.9529340267181396

Для даних про продаж авокадо в якості тестових даних були взяті дані з 2015 по 2018 рік. Так як моделі ARMA і ARIMA не призначені для аналізу даних які мають тренд і сезонність, результати роботи цих алгоритмів на двох наборах даних підтвердили їх неефективність, для передбачення зростання цін на авокадо було вибрано моделі SARIMA і Холта-Вінтерса.

Графіки прогнозів представлені на рисунку 3.41. Таблиця 3.2 містить інформацію про MSE, RMSE і часу роботи кожного з алгоритмів.

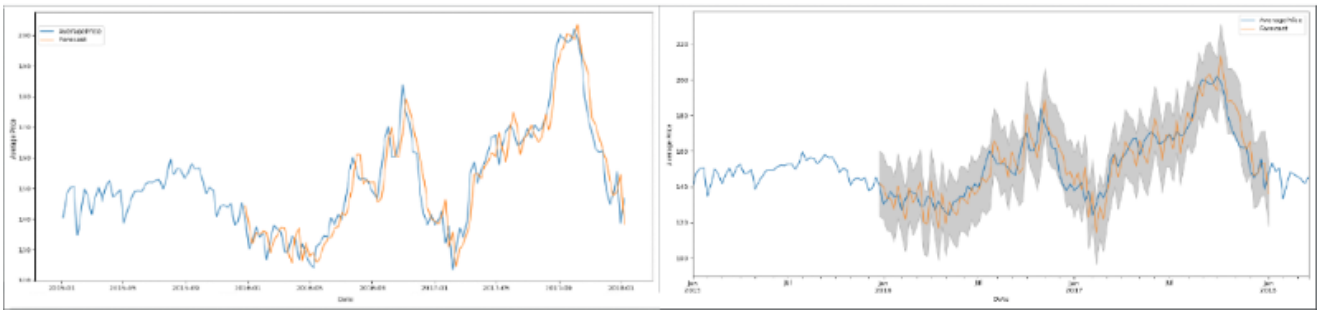


Рисунок 3.41 – Зліва результати прогнозу Холтом-Вінтерсом, праворуч SARIMA

Таблиця 3.2 – Результати помилок та час роботи для моделей прогнозування SARIMA та Холта–Вінтерса

Модель	MSE	RMSE	Час роботи, сек
SARIMA	81.21933549380245	9.012177067379582	21.364629983901978
Holt-Winters	42.05946680058604	6.485327038830504	2.7933509349823

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об’єкту, що мають вплив на персонал

У даному дипломному проекті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до ДСН 3.3.6.042-99 [41] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу

слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до НПАОП 0.00-7.15-18 [42], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

– підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
– підвищений рівень статичної електрики;
– підвищений рівень електромагнітних випромінювань;
– підвищена або знижена температура повітря робочої зони;
– підвищений або знижений рух повітря;
– підвищена або знижена вологість повітря;
– відсутність або недостатність природного світла;
– підвищена пульсація світлового потоку;
– недостатня освітленість робочого місця;
– підвищений рівень шуму на робочому місці;
– розумове перенапруження;
– емоційні навантаження;
– монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

– значення сили струму;
– електричного опору тіла людини і тривалості протікання через нього струму;
– роду і частоти струму;

– індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

– заземлення електроустановок;

– занулення;

– захисне відключення;

– електричне розділення мережі;

– використовування малої напруги;

– ізоляція частин, що проводять струм;

– огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_K = \frac{\text{Uф}}{\text{Zп} + \frac{\text{Zт}}{3}}, \qquad (4.1)$$

де Uф - номінальна фазна напруга мережі, В;

Zп - повний опір петлі, створене фазними і нульовими дротами, Ом;

ZT - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [43]: Zт /3 = 0,1 Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу (4.2.) :

$$\text{Zп} = \sqrt{\text{Rп}^2 + \text{Xп}^2}, \qquad (4.2)$$

де Rп = Rф + Ro - сумарний активний опір фазного Rф і нульового Ro дротів, Ом;

Xп - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту S = 2.5 мм, тоді згідно таблицям 5 і 6 [43], має такий опір:

Хп = 0,11 Ом;

Rф = 7,55 Ом;

Rо = 7,55 Ом.

Отже, Rп = 7,55 + 7,55 = 15,1 Ом.

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Zп = \sqrt{15{,}1^2 + 0{,}11^2} \approx 15{,}1 \text{ (Ом).}$$

Струм однофазного короткого замикання рівний:

$$Iк = \frac{220}{15{,}1 + 0{,}1} = 14{,}47 \text{ (А).}$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$Iк \geq k * Iн , \quad (4.3)$$

де Iн - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму Iн, А.

Коефіцієнт кратності нелінійного струму Iн розраховується по формулі (4.4.) :

$$Iн = P / U , \quad (4.4)$$

де P = 220 Вт - споживана потужність;

U = 220 В - робоча напруга;

k = 3 А - для плавких вставок.

Отже, Iн = 220 / 220 = 1 А.

Підставивши значення у вираз (4.3), одержимо:

14,47> 3*1.

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються ДСН 3.3.6.042-99[41], ДБН, відповідними ГОСТами і ОСТамі з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводити до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до ДСН 3.3.6.042-99 [41] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні . За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу(категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22…24	40…60	0,1
Тепло	23…25	40…60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми

Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в ДБН (30 $м^3$ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплену).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту TCO'03.

TCO'03 пред’являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [44].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до ДБН В.2.5-28-2006 [47] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки a=12 м, шириною b=10 м, заввишки з=4 м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k)/(F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м2;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використовування освітлювальної установки = 0,6;

M - кількість ламп у світильнику =2.

З формули (4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k)/(F \cdot U \cdot M), \quad (4.6)$$

Отже, $n = (200 \cdot 120 \cdot 1{,}2 \cdot 1{,}5)/(3120 \cdot 0{,}6 \cdot 2) = 12$

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

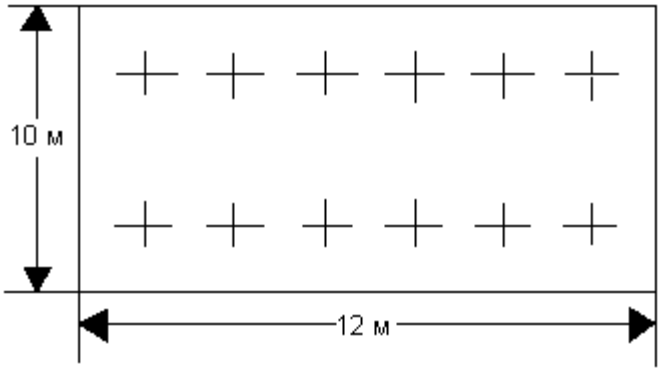


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов’язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

– поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;

– полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

– стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

– пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

– деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно ДСТУ Б В.1.1-36-2016 [48] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір

усередині приміщення відноситься до вогненебезпечної зони класу П - IIа (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

– іскри при замиканні і розмиканні ланцюгів;

– іскри і дуги коротких замикань;

– перегріви від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеін, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідрідні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об’єктів народного господарства регламентується і забезпечується системами запобігання пожежам і протипожежному захисту[48]. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

– застосування устаткування, що задовольняє вимогам електростатичної безпеки;

– застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;

– виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;

– підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з пальним середовищем, нижче гранично допустимої, становить 80% як найменшої температури самозаймання пального.

– заміна небезпечних технологічних операцій більш безпечними;

– ізольоване розташування небезпечних технологічних установок і устаткування;

– зменшення кількості пальних і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;

– запобігання можливості утворення пальних сумішей на лінії, вентиляційних системах і ін.;

– механізація, автоматизація та справність(потокова) виробництва;

– суворе дотримання стандартів і точне виконання встановленого технологічного режиму;

– запобігання можливості появи в небезпечних місцях джерел запалення;

– запобігання розповсюдженню пожеж і вибухів;

– використовування устаткування і пристроїв, при роботі яких не виникає джерел запалення;

– виконання вимог сумісного зберігання речовин і матеріалів;

– наявність громовідводу;

– ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

– електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолоджування і кондиціонування;

– система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;

– робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, ломи, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використовування промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

4.5 Вплив на навколишнє природнє середовище

Діяльність за темою магістерської роботи, а саме: дослідження ефективності використання засобів векторної обробки ядер сучасних процесорів в процесі її виконання впливає на навколишнє природнє середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища»[51], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення»[53], Законом України «Про відходи»[52].

В процесі діяльності виконанням дипломного проектування виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- відпрацьовані люмінесцентні лампи - I клас небезпеки.
- змінні носії інформації - IV клас небезпеки.
- відпрацьовані вогнегасники - IV клас небезпеки.
- макулатура - IV клас небезпеки

ВИСНОВКИ

У рамках магістерської роботи було розроблено і досліджено методи прогнозування часових рядів а також комп'ютерна програма, яка дозволяє користувачам входити до CRM системи під своїми логіном та паролем та обравши таблиці для аналізу зробити прогноз, також є можливість загружати csv файли з комп'ютера для аналізу та прогнозування. У рамках дослідження було протестовано моделі ARMA, ARIMA, SARIMA та Холта-Вінтерса. Для дослідження було обрано три набори даних:

– щомісячні дані про продаж автомобілів в період з 2007 до 2017 року, які мають 2694 записи;

– щомісячні дані про продаж автомобілів в період з 2016 по 2018 рік, які мають 36 записів;

– щотижневі дані про ціну авокадо в період з 2015 по 2018 рік, які мають 18249 записів.

Такий вибір даних обумовлено тим, що вони мають різну кількість та різну щільність записів, що впливає на точність прогнозу. Наприклад, дані шодо росту цін на авокадо мають щотижневі записи, у відмінності від двох інших наборів де записи тільки щомісячні. Тобто за 1 місяць росту цін часовий ряд має більш детальну інформацію ніж за 1 місяць продажу автомобілів. Два набори даних з продажу автомобілів були обрані для того, щоб показати як детальність та кількість записів може вплинути на прогноз. Також продаж автомобілів за 2 роки потребує менше часу на опрацювання алгоритмом, тобто це дозволяє порівняти час роботи.

Після усіх тестів можна зробити висновок, що алгоритм потрійного експоненційного згладжування працює ефективніше та швидше, ніж SARIMA. На всіх наборах даних Холт-Вінтерс продемонстрував кращий результат. Результати методу Холта-Вінтерса мають кращу точність, але похибка SARIMA не сильно більше.

На датасеті з продажу авокадо результат потрійного експоненційного згладжування був кращий ніж всі інші, та прогноз практично співпадав з реальними даними. Самий гірший результат був на наборі даних з продажу автомобілів за 2 роки, через те що інформації мало та вона не щотижнева, тобто алгоритм не має достатньо інформації для побудови реалістичного прогнозу.

Результати порівняння усіх моделей на одному наборі даних підтвердили, що обирати модель до конкретних даних є важливою умовою прогнозування, тому що моделі

ARMA та ARIMA показали найгірший результат, через те, що вони не є оптимальними для такого набору даних.

Також, через те, що Холт-Вінтерс не обчислює авторегресію та не має багато параметрів для вибору на всіх тестах він має кращий результат з точки зору часу роботи. Навіть ARMA, яка потребує лище 2 параметри та не робить жодних додаткових операцій працює довше, ніж Холт-Вінтерс.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці. А також визначені основні екологічні аспекти впливу на навколишнє природнє середовище та зазначені заходи щодо поводження з ними.

Результати атестаційної роботи були апробовані на міжнародній конференцій «Майбутній науковець 2020».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1) Hamilton, J. D. (1994). Time series analysis (Vol. 2, pp. 690-696). Princeton, NJ: Princeton university press.

2) Wei, W. W. (2006). Time series analysis. In The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2.

3) Cryer, J. D., & Chan, K. S. (2008). Time series analysis. With applications in R. Springer.

4) Fuller, W. A. (2009). Introduction to statistical time series (Vol. 428). John Wiley & Sons.

5) Chatfield, C. (2000). Time-series forecasting. Chapman and Hall/CRC.

6) Taylor, S. J. (2008). Modelling financial time series. world scientific.

7) Hurvich, C. M., & Tsai, C. L. (1989). Regression and time series model selection in small samples. Biometrika, 76(2), 297-307.

8) Chatfield, C., & Yar, M. (1988). Holt-Winters forecasting: some practical issues. Journal of the Royal Statistical Society: Series D (The Statistician), 37(2), 129-140.

9) Chatfield, C. (1978). The Holt-winters forecasting procedure. Journal of the Royal Statistical Society: Series C (Applied Statistics), 27(3), 264-279.

10) Gelper, S., Fried, R., & Croux, C. (2010). Robust forecasting with exponential and Holt–Winters smoothing. Journal of forecasting, 29(3), 285-300.

11) Hillmer, S. C., & Tiao, G. C. (1982). An ARIMA-model-based approach to seasonal adjustment. Journal of the American Statistical Association, 77(377), 63-70.

12) Ediger, V. Ş., & Akar, S. (2007). ARIMA forecasting of primary energy demand by fuel in Turkey. Energy policy, 35(3), 1701-1708.

13) Mantula, E. V., & Mashtalir, S. V. (2013). Method of Adaptive Forecasting Based On Multidimensional Linear Extrapolation. International Journal of Research in Engineering and Science, 1(4), 31-37.

14) Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In Proceedings of the 9th International Conference on Information Management and Engineering (pp. 60-63). ACM.

15) Мантула, Е., & Машталир, С. (2014). Adaptive polynomial neuronetwork predicting model of time series and its training. Eastern-European Journal Of Enterprise Technologies, 2(4(68), 16.

16) Богучарский, С. И., Машталир, С. В., & Столбовой, М. И. (2018). Быстрое обнаружение изменения свойств многомерных временных рядов на основе идентификационного подхода к ансамблю моделей.

17) Kinoshenko, D., Mashtalir, S., Stephan, A., & Vinarski, V. (1993). Neural Network Segmentation Of Video Via Time Series Analysis. INFORMATION THEORIES & APPLICATIONS, 232.

18) Damodar, N. G. (2004). Basic econometrics. The McGraw− Hill Companies.

19) Contreras, J., Espinola, R., Nogales, F. J., & Conejo, A. J. (2003). ARIMA models to predict next-day electricity prices. IEEE transactions on power systems, 18(3), 1014-1020.

20) Pappas, S. S., Ekonomou, L., Karamousantas, D. C., Chatzarakis, G. E., Katsikas, S. K., & Liatsis, P. (2008). Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models. Energy, 33(9), 1353-1360.

21) Chen, J. F., Wang, W. M., & Huang, C. M. (1995). Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. Electric Power Systems Research, 34(3), 187-196.

22) Hyndman, R., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). Forecasting with exponential smoothing: the state space approach. Springer Science & Business Media.

23) Kalekar, P. S. (2004). Time series forecasting using holt-winters exponential smoothing. Kanwal Rekhi School of Information Technology, 4329008(13).

24) Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.

25) De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. International journal of forecasting, 22(3), 443-473.

26) Quenouille, M. H., & Quenouille, M. H. (1957). The analysis of multiple time-series (Vol. 1). London: Griffin.

27) Ostrom, C. W. (1990). Time series analysis: Regression techniques (No. 9). Sage.

28) Schwartz, J., & Marcus, A. (1990). Mortality and air pollution j london: a time series analysis. American journal of epidemiology, 131(1), 185-194.

29) Hargreaves, C. (1994). Non-stationary time series analysis and cointegration. Oxford University Press.

30) Ahmed, S. R., & Cook, A. R. (1982). Application of time-series analysis techniques to freeway incident detection. Transportation Research Record, 841, 19-21.

31) Ahmed, M. S., & Cook, A. R. (1979). Analysis of freeway traffic time-series data by using Box-Jenkins techniques (No. 722).

32) Naidu, P. S. (1995). Modern spectrum analysis of time Series: Fast algorithms and error control techniques. CRC Press.

33) Valenzuela, O., Rojas, I., Rojas, F., Pomares, H., Herrera, L. J., Guillén, A., ... & Pasadas, M. (2008). Hybridization of intelligent techniques and ARIMA models for time series prediction. Fuzzy sets and systems, 159(7), 821-845.

34) Athiyaman, A., & Robertson, R. W. (1992). Time series forecasting techniques: Short-term planning in tourism. International Journal of Contemporary Hospitality Management, 4(4).

35) Ljung, G. M., & Box, G. E. (1978). On a measure of lack of fit in time series models. Biometrika, 65(2), 297-303.

36) Harvey, A. C. (1990). Forecasting, structural time series models and the Kalman filter. Cambridge university press.

37) Sowell, F. (1992). Maximum likelihood estimation of stationary univariate fractionally integrated time series models. Journal of econometrics, 53(1-3), 165-188.

38) Basu, S., Mukherjee, A., & Klivansky, S. (1996, March). Time series models for internet traffic. In Proceedings of IEEE INFOCOM'96. Conference on Computer Communications (Vol. 2, pp. 611-620). IEEE.

39) Hasza, D. P., & Fuller, W. A. (1982). Testing for nonstationary parameter specifications in seasonal time series models. The Annals of Statistics, 1209-1216.

40) Chen, C., & Tiao, G. C. (1990). Random level-shift time series models, ARIMA approximations, and level-shift detection. Journal of Business & Economic Statistics, 8(1), 83-97.

41) ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих приміщень. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99

42) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров`я працівників під час роботи з екранними пристроями. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18

43) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та

44) Номенклатура видів захисту. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/ru/v0037831-11

45) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98

46) ДБН В.2.5-67:2013. Опалення вентиляція та кондиціонування. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0024858-13

47) ДБН В.2.5-28-2006. Природне і штучне освітлення. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0168667-06

48) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419

49) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/z0379-96

50) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.

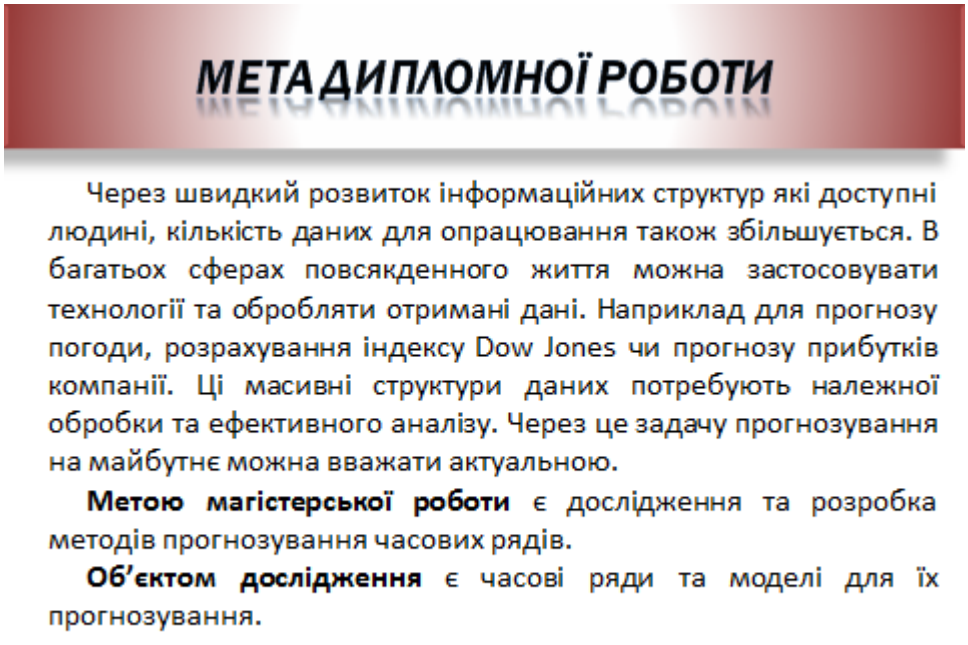
51) Закон України «Про охорону навколишнього природного середовища» . Вводиться в дію Постановою ВР № 1268-XII від 26.06.91, ВВР, 1991, № 41, ст.547. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/1264-12

52) Закон України «Про відходи». Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст.242. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр

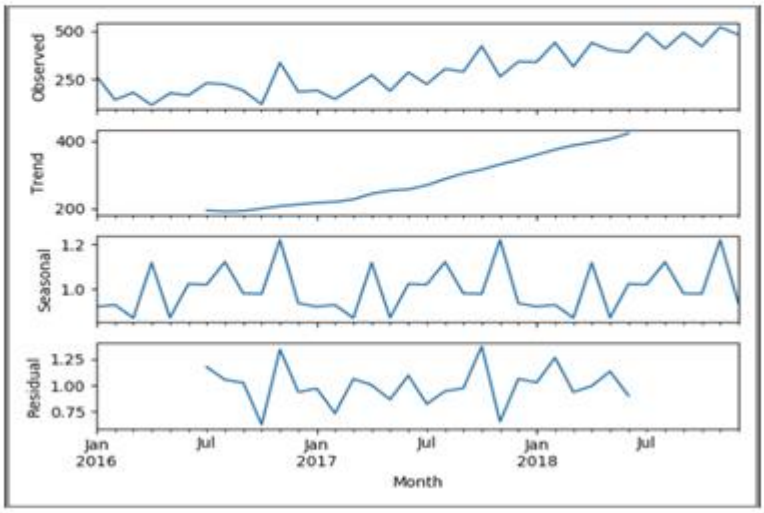
53) Закон України «Про забезпечення санітарного та епідемічного благополуччя населення». Відомості Верховної Ради України (ВВР), 1994, № 27, ст.218. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/4004-12

ДОДАТОК А.

Електронні плакати



ОСНОВНІ КОМПОНЕНТИ ЧАСОВОГО РЯДУ



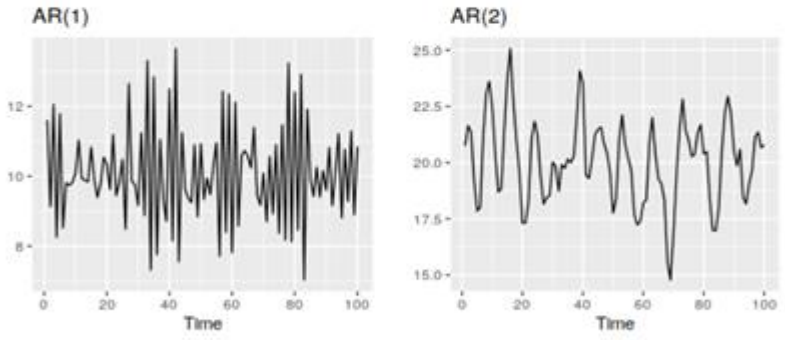
СФЕРИ ЗАСТОСУВАННЯ ЧАСОВИХ РЯДІВ

МОДЕЛІ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1) Модель авторегресії та ковзного середнього – ARMA
   - Авторегресійна модель AR(p)
   - Модель ковзного середнього MA(q)
2) Інтегрована модель авторегресії та ковзного середнього ARIMA(p,d,q)
3) Сезонна інтегрована модель авторегресії SARIMA

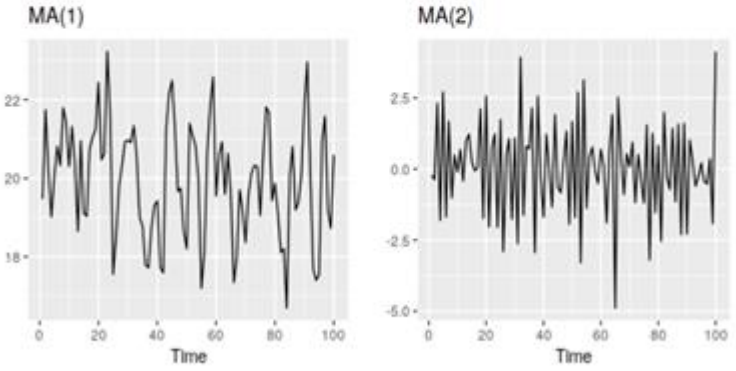
АВТОРЕГРЕСІЙНА МОДЕЛЬ AR(P)

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \Phi_p y_{t-p} + \varepsilon_t \quad ,$$



МОДЕЛЬ КОВЗНОГО СЕРЕДНЬОГО MA(Q)

$$y_t = c+\varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \dots + \theta_q\varepsilon_{t-q},$$



ІНТЕГРОВАНА МОДЕЛЬ АВТОРЕГРЕСІЇ ТА КОВЗНОГО СЕРЕДНЬОГО ARIMA(P,D,Q)

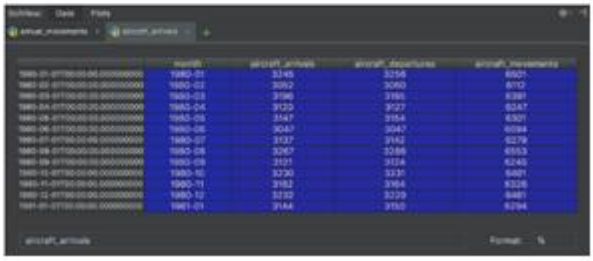
$$y'_t = c + \Phi_1 y'_{t-1} + \dots + \Phi_p y'_{t-p} + \theta_1\varepsilon_{t-1} + \dots + \theta_q\varepsilon_{t-q} + \varepsilon_t,$$

СЕЗОННА ІНТЕГРОВАНА МОДЕЛЬ АВТОРЕГРЕСІЇ SARIMA

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

$$\varphi_P(L^m)\phi(L)\nabla_m^D\nabla^d y_t = \Theta_Q(L^m)\theta(L)\varepsilon_t\,,$$

ВИБОР СЕРЕДОВИЩА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ



ПРОГРАМНА РЕАЛІЗАЦІЯ

```
def prediction(myList = [], *args):
    combineQuery = 'SELECT '

    for i, val in enumerate(myList):
        print(myList[i])
        combineQuery += myList[i]
        if i < len(myList)-1:
            combineQuery += ','


    sf_select = sf.query_all((combineQuery + ' FROM ' + globalobjectName))
    sf_df = pd.DataFrame(sf_select['records']).drop(columns='attributes')

    for i, name in enumerate(list(sf_df)):
        if name.endswith('__c'):
            print(name[:len(name)-3])
            sf_df = sf_df.rename(columns={name: name[:len(name)-3]})
```

```
def query_all(self, query, include_deleted=False, **kwargs):
    """
    Arguments

    * query -- the SOQL query to send to Salesforce, e.g.
                SELECT Id FROM Lead WHERE Email = "waldo@somewhere.com"
    * include_deleted -- True if the query should include deleted records.
    """
    result = self.query(query, include_deleted=include_deleted, **kwargs)
    all_records = []

    while True:
        all_records.extend(result['records'])
        # fetch next batch if we're not done else break out of loop
        if not result['done']:
            result = self.query_more(result['nextRecordsUrl'],
                                     identifier_is_url=True)
        else:
            break

    result['records'] = all_records
    return result
```

ПРОГРАМНА РЕАЛІЗАЦІЯ

```
result = adfuller(training_data['Quantity'])
print('ADF Statistic: {}'.format(result[0]))
print('p-value: {}'.format(result[1]))
print('Critical Values:')
for key, value in result[4].items():
    print('\t{}: {}'.format(key, value))
```

```
    xdall = lagmat(xdiff[:, None], maxlag, trim='both', original='in')
    nobs = xdall.shape[0]
    xdall[:, 0] = xo[-nobs-1:-1]  # replace 0 xdiff with level of x
    xdshort = xdiff[-nobs:]
    usedlag = maxlag
else:
    usedlag = maxlag
    icbest = None
if regression != 'nc':
    resols = OLS(xdshort, add_trend(xdall[:, :usedlag + 1],
                 regression)).fit()
else:
    resols = OLS(xdshort, xdall[:, :usedlag + 1]).fit()

adfstat = resols.tvalues[0]


# Get approx p-value and critical values
pvalue = mackinnonp(adfstat, regression=regression, N=1)
critvalues = mackinnoncrit(N=1, regression=regression, nobs=nobs)
critvalues = {"1%" : critvalues[0], "5%" : critvalues[1],
              "10%" : critvalues[2]}
if store:
    resstore.resols = resols
    resstore.maxlag = maxlag
    resstore.usedlag = usedlag
    resstore.adfstat = adfstat
    resstore.critvalues = critvalues
    resstore.nobs = nobs
    resstore.H0 = ("The coefficient on the lagged level equals 1 - "
                   "unit root")
    resstore.HA = "The coefficient on the lagged level < 1 - stationary"
    resstore.icbest = icbest
    resstore._str = 'Augmented Dickey-Fuller Test Results'
    return adfstat, pvalue, critvalues, resstore
else:
    if not autolag:
        return adfstat, pvalue, usedlag, nobs, critvalues
```

ПРОГРАМНА РЕАЛІЗАЦІЯ

```
acfDIFF = plot_acf(training_data, lags=20, title='ACF', alpha=0.5) #1
acfDIFF.show()
pacfDIFF = plot_pacf(training_data, lags=20, title='PACF', alpha=0.5) #1
pacfDIFF.show()
```

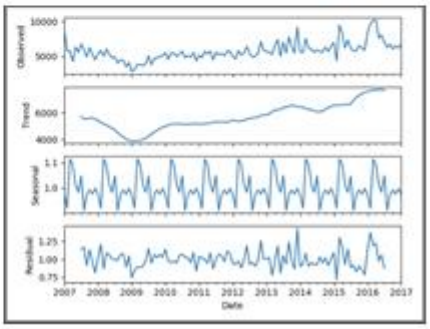
```
y_forecasted = model.predict(start = '2014-01-01', end = '2017-01-01', dynamic=False)
print(y_forecasted)
print(carSalesData['2014-01-01': '2017-01-01'])
actual_data = carSalesData['2014-01-01': '2017-01-01']
y_truth = actual_data
mse = mean_squared_error(y_truth, y_forecasted)
print(mse)
print(np.sqrt(mse))
```

ПРОГРАМНА РЕАЛІЗАЦІЯ

```
AIC_list = pd.DataFrame([], columns=['param','param_seasonal','AIC'])
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(training_data,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)

            results = mod.fit()

            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
            temp = pd.DataFrame([[param, param_seasonal, results.aic]], columns=['param','param_seasonal','AIC'])
            AIC_list = AIC_list.append(temp, ignore_index=True)
            del temp
        except:
            continue

m = np.nanmin(AIC_list['AIC'].values) # Find minimum value in AIC
i = AIC_list['AIC'].tolist().index(m) # Find index number for lowest AIC
Min_AIC_list = AIC_list.iloc[i,:]
```

```
model = ExponentialSmoothing(training_data, seasonal='mul', seasonal_periods=12).fit(use_brute=True)
pred = model.predict(start='2014-01-01', end='2017-01-01')
```

ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ПРОГНОЗУВАННЯ



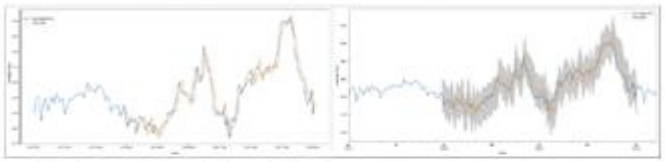
ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ПРОГНОЗУВАННЯ

	param	param_seasonal	AIC
0	(0, 0, 0)	(0, 0, 0, 12)	2174.831548
1	(0, 0, 0)	(0, 1, 0, 12)	1644.961198
2	(0, 0, 0)	(0, 2, 0, 12)	1523.414835
3	(0, 0, 0)	(1, 0, 0, 12)	1669.421807
4	(0, 0, 0)	(1, 1, 0, 12)	1449.976032
5	(0, 0, 0)	(1, 2, 0, 12)	1287.150869
6	(0, 0, 0)	(2, 0, 0, 12)	1449.239866
7	(0, 0, 0)	(2, 1, 0, 12)	1235.993981
8	(0, 0, 0)	(2, 2, 0, 12)	1048.794666
9	(0, 1, 0)	(0, 0, 0, 12)	1823.545620
10	(0, 1, 0)	(0, 1, 0, 12)	1669.854199
11	(0, 1, 0)	(0, 2, 0, 12)	1545.654233
12	(0, 1, 0)	(1, 0, 0, 12)	1638.927858
13	(0, 1, 0)	(1, 1, 0, 12)	1463.822420
14	(0, 1, 0)	(1, 2, 0, 12)	1301.882847
15	(0, 1, 0)	(2, 0, 0, 12)	1443.122257
16	(0, 1, 0)	(2, 1, 0, 12)	1254.177343
17	(0, 1, 0)	(2, 2, 0, 12)	1085.758858
18	(0, 2, 0)	(0, 0, 0, 12)	1924.724938
19	(0, 2, 0)	(0, 1, 0, 12)	1761.753854

2016-01-01	7072.129884
2016-02-01	4353.863958
2016-03-01	10013.709294
2016-04-01	11176.027404
2016-05-01	9089.441422
2016-06-01	7524.236831
2016-07-01	7663.204082
2016-08-01	7325.149893
2016-09-01	6831.290121
2016-10-01	7452.484623
2016-11-01	5786.740922
2016-12-01	6207.394055
2017-01-01	8978.169191

2016-01-01	8441
2016-02-01	9628
2016-03-01	10211
2016-04-01	10231
2016-05-01	7637
2016-06-01	8045
2016-07-01	7077
2016-08-01	6299
2016-09-01	6724
2016-10-01	6041
2016-11-01	6512
2016-12-01	6178
2017-01-01	6834

ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ПРОГНОЗУВАННЯ



Модель	MSE	RMSE	Час роботи, сек
SARIMA	81.21933549380245	9.012177067379582	21.36
Holt-Winters	42.05946680058604	6.485327038830504	2.79

ВИСНОВКИ

У рамках магістерської роботи було розроблено і досліджено методи прогнозування часових рядів а також комп'ютерна програма, яка дозволяє користувачам входити до CRM системи під своїми логіном та паролем та обравши таблиці для аналізу зробити прогноз, також є можливість загружати csv файли з комп'ютера для аналізу та прогнозування. У рамках дослідження було протестовано моделі ARMA, ARIMA, SARIMA та Холта-Вінтерса.

Після усіх тестів можна зробити висновок, що алгоритм потрійного експоненційного згладжування працює ефективніше та швидше, ніж SARIMA. На всіх наборах даних Холт-Вінтерс продемонстрував кращий результат. Результати методу Холта-Вінтерса мають кращу точність, але похибка SARIMA не сильно більше.

Також, через те, що Холт-Вінтерс не обчислює авторегресію та не має багато параметрів для вибору на всіх тестах він має кращий результат з точки зору часу роботи. Навіть ARMA, яка потребує лище 2 параметри та не робить жодних додаткових операцій працює довше, ніж Холт-Вінтерс.