Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: **«ПОРІВНЯННЯ АВТОРЕГРЕСИВНОЇ МОДЕЛІ З МЕТОДОМ ЕКСПОНЕНЦІАЛЬНОГО ЗГЛАДЖУВАННЯ ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВОГО РЯДУ»**

Виконав: студент 4-го року навчання
освітньої програми «Прикладна математика»,
спеціальності 113 Прикладна математика

Пархомчук Олександр Павлович

Керівник: Дрінь С. С.,
кандидат фіз.-мат. наук, ст. викладач

Рецензент: Жуковська О.А

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____
«____» _____ 20____ р.

Київ – 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
проф., д.ф.-м.н.
_____М. М. Глибовець
(підпис)
„____”_____2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну роботу

студенту Пархомчук Олександру Павловичу факультету інформатики 4 курсу
ТЕМА Порівняння авторегресивної моделі з методом експоненціального згладжування для прогнозування часових рядів Зміст ТЧ до кваліфікаційної роботи:

Індивідуальне завдання
Вступ
1 Теоретичні відомості з авторегресивних моделей
2 Теоретичні відомості зі експоненціального згладжування
3 Порівння авторегресивних моделей з методом експонеціального згладжування у прогнозуванні часових рядів на основі реальних даних
4 Розробка комп’ютерної програми
Висновки
Список літератури
Додатки

Дата видачі „23” жовтня 2021 р. Керівник _____
(підпис)
Завдання отримав _____
(підпис)

Тема: Порівняння авторегресивної моделі з методом експоненціального згладжування для прогнозу часового ряду

Календарний план виконання кваліфікаційної роботи:

№ п/п	Назва етапу кваліфікаційної роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на кваліфікаційну роботу.	23.10.2021	
2.	Огляд технічної літератури за темою роботи.	12.11.2021	
3.	Опрацювання матеріалів	6.02.2022	
4.	Погодження плану робіт з керівником	1.04.2022	
5.	Розв’язання поставленої задачі, аналіз проблеми	5.05.2022	
6.	Розробка комп’ютерної реалізації алгоритму	19.05.2022	
7.	Виконання порівняльного аналізу результатів прогнозування	22.05.2022	
8.	Створення слайдів та написання доповіді	9.06.2022	
9.	Остаточне оформлення роботи	17.06.2022	
10.	Захист кваліфікаційної роботи	4.07.2022	

Студент: Пархомчук О.П.

Керівник: Дрінь С.С.

“23” жовтня 2021 р

ЗМІСТ

Анотація

Задачею роботи, є порівняння авторегресивної моделі з методом експоненціального згладжування для прогнозування часового ряду. Теоретична частина присвячена основним теоріям для прогнозування часового ряду, які будуть використовуватися, а саме, авторегресивні моделі та метод експоненціального згладжування. Ключовими словами є: AR, ARIMA, TES, DES. Для вирішення задачі використовувалась мова програмування Python. Отримані результати показали, що найкращий результат буде досягатися при використанні методу експоненціального згладжування.

ВСТУП

Прогнозування часових рядів є одним із найбільш застосовуваних методів науки про дані в бізнесі, фінансах, управлінні ланцюгами поставок, плануванні виробництва та запасів. Багато проблем прогнозування включають часовий компонент і, таким чином, вимагають екстраполяції даних часових рядів або прогнозування часових рядів. Прогнозування передбачає використання моделей, що відповідають історичним даним, і використання їх для прогнозування майбутніх спостережень.

Прогнозування часових рядів означає прогнозування або передбачення майбутнього за певний період часу. Це тягне за собою розробку моделей на основі попередніх часових періодів та їх використання для майбутніх стратегічних рішень.

Прогнозування та оцінка майбутніх часових періодів здійснюється на основі того, що вже сталося. Часовий ряд додає залежність від часового параметру у спостереженнях. Часовий параметр є одночасно і обмеженням, і структурою, яка забезпечує джерело додаткової інформації. Перш ніж обговорювати методи прогнозування часових рядів, давайте визначимо прогнозування часових рядів більш детально.

Прогнозування часових рядів — це модель для передбачення подій через елементи попередніх періодів. Він передбачає майбутні події, аналізуючи тенденції минулого, виходячи з припущення, що майбутні тенденції будуть подібними до історичних тенденцій. Він використовується в багатьох областях дослідження в різних областях, включаючи: астрономія, бізнес-планування, інженерія управління, прогноз землетрусу, економетрія, фінансова математика, розпізнавання образів, розподіл ресурсів, обробка сигналу, статистика, прогноз погоди.

Прогнозування часових рядів починається з аналізу часового ряду. Аналітики вивчають історичні дані та перевіряють закономірності

декомпозиції за часом, такі як тенденції, сезонні закономірності, циклічні закономірності та регулярність. Багато сфер, включаючи маркетинг, фінанси та продажі, використовують певну форму прогнозування часових рядів для оцінки ймовірних технічних витрат та споживчого попиту. Моделі для даних часових рядів можуть мати багато форм і представляти різні стохастичні процеси.

Метою цього дослідження є розглянути методи прогнозування часових рядів і коротко пояснити роботу методів прогнозування часових рядів. Ми обговоримо часові ряди, методи, які використовуються в прогнозуванні часових рядів, переваги та недоліки прогнозування часових рядів. Ми також обговоримо підходи та застосування різних методів, що використовуються в прогнозуванні часових рядів. Мета — порівняти авторегресивну модель з методом експоненціального згладжування для прогнозування часового ряду**.** Дані аналізуються для отримання статистичної інформації, характеристик даних і прогнозування результатів. Оскільки дані можуть мати тенденцію відповідати шаблону в даних часових рядів, моделі машинного навчання важко прогнозувати належним чином, тому аналіз часових рядів і його підходи спрощують прогнозування.

Перелік прийнятих скорочень

ACF – autocorrelation function

PACF – partial autocorrelation function

OLS - ordinary least squares

Предиктори - структурно організована система, функцією якої є прогнозування, тобто незалежна змінна

SES – Simple Exponential Smoothing

DES – Double Exponential Smoothing

TES – Triple Exponential Smoothing

РОЗДІЛ 1. Огляд існуючих теорій з цього питання

1.1 AR моделі

Часовий ряд — це послідовність елементів певної змінної, які зроблені за певний проміжок часу. Зазвичай вимірювання проводяться в рівномірний час - наприклад, щодня, щомісяця, щороку та інші. Давайте спочатку розглянемо задачу, в якій маємо y-змінну, як часовий ряд. Як приклад, ми можемо мати y міру глобальної температури, за вимірюваннями, які спостерігаються щороку. Щоб підкреслити, що ми вимірювали значення протягом певного часу, ми використовуємо «t» як індекс, а не звичайне «i», тобто y_t означає вимірювання y в періоді часу t[3, 4, 5].

Авторегресивні моделі діють на основі того, що минулі значення впливають на поточні, що робить статистичний метод популярним для аналізу природи, економіки та інших процесів, які змінюються з часом. Множинні регресійні моделі прогнозують змінну, використовуючи лінійну комбінацію предикторів(потрібно уточнити раніше), тоді як авторегресивні моделі використовують комбінацію попередніх значень змінної про що буде вказано згодом[3, 4, 5].

Авторегресивний процес $AR(1)$ — це процес, у якому поточне значення засноване на попередньому значенні, тоді як процес $AR(2)$ — це процес, у якому поточне значення засноване на двох попередніх значеннях. Процес $AR(0)$ використовується для білого шуму і не має залежності від попередніх елементів. На додаток до цих авторегресивних процесів існує також багато різних способів обчислення коефіцієнтів, які використовуються в розрахунках, наприклад узагальнений метод найменших квадратів[3, 4, 5].

Авторегресивна модель – це коли значення часового ряду залежать від попередніх значеннь з того самого ряду. Наприклад , y_t на y_{t-1}:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t \quad (1.1)$$

У цій регресійній моделі залежна змінна за попередній період часу стала предиктором. Порядок авторегресії — це кількість значень, що безпосередньо передують у серії, які використовуються для прогнозування значення в даний момент. Отже, попередня модель — це авторегресія першого порядку, записана як $AR(1)$.

Якщо ми хочемо передбачити y цей рік (y_t) за допомогою вимірювань глобальної температури за попередні два роки (y_{t-1}, y_{t-2}), то авторегресивна модель для цього може бути:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \varepsilon_t \quad (1.2)$$

Ця модель є авторегресією другого порядку, записаною як $AR(2)$, оскільки значення в момент часу t прогнозується на основі значень $t-1$ та $t-2$. У більш загальному вигляді авторегресія k - порядку, записана як $AR(k)$, є множинною лінійною регресією, в якій значення ряду в будь-який

момент часу t є (лінійною) функцією значень у моменти часу $t-1, t-2, \ldots, t-k$.

Ці концепції та методи використовуються технічними аналітиками для прогнозування цін на цінні папери. Однак, оскільки авторегресивні моделі базують свої прогнози лише на минулій інформації, вони неявно припускають, що фундаментальні сили, які вплинули на минулі ціни, не зміняться з часом. Це може призвести до дивовижних і неточних прогнозів, якщо основні сили, про які йде мова, насправді змінюються, наприклад, якщо галузь зазнає швидких і безпрецедентних технологічних перетворень[3, 4, 5].

Тим не менш, трейдери продовжують удосконалювати використання авторегресивних моделей для цілей прогнозування. Чудовим прикладом є авторегресивне інтегроване ковзне середнє ($ARIMA$), складна модель авторегресії, яка може враховувати тенденції, цикли, сезонність, помилки та інші нестатичні типи даних при складанні прогнозів[3, 4, 5].

Коефіцієнт кореляції між двома значеннями в часовому ряді називається функцією автокореляції (ACF). Наприклад, ACF для часового ряду y_t визначається як:

$$Corr(y_t, y_{t-k}) \quad (1.3)$$

Це значення k є часовим розривом, який розглядається, і називається лагом. Автокореляція з відставанням 1 (тобто $k = 1$ у наведеному вище) – це кореляція між значеннями, які знаходяться на відстані одного часового періоду. Загалом, автокореляція з затримкою k — це кореляція між значеннями, які знаходяться на відстані k часових періодів.

ACF – це спосіб вимірювання лінійного співвідношення між спостереженням у момент t і спостереженнями в попередні моменти. Якщо ми припустимо модель $AR(k)$, тоді ми можемо захотіти лише виміряти зв'язок між y_t та y_{t-k} та відфільтрувати лінійний вплив випадкових величин, які лежать між ними (тобто, $y_{t-1}, y_{t-2}, \ldots, y_{t-k}$), що вимагає перетворення часового ряду. Тоді шляхом розрахунку кореляції перетвореного часового ряду отримуємо часткову автокореляційну функцію (PACF) [5].

PACF найбільш корисний для визначення порядку авторегресивної моделі. Зокрема, вибіркові часткові автокореляції, які значно відрізняються від 0, вказують на відставання змінної y, тобто вважаються факторами y_t.

Графічні підходи до оцінки відставання авторегресивної моделі включають перегляд значень ACF і PACF у порівнянні з лагом. Якщо ви бачите великі значення ACF і невипадковий шаблон, то, ймовірно, значення послідовно корелюють. На графіку залежності PACF від запізнення шаблон зазвичай буде виглядати випадковим, але великі значення PACF при заданому відставанні вказують на це значення як можливий вибір для порядку авторегресивної моделі. Важливо, щоб вибір порядку мав сенс. Наприклад, припустимо, що у вас є показники артеріального тиску за кожен день протягом останніх двох років. Ви можете виявити, що модель AR(1) або AR(2) підходить для моделювання артеріального тиску. Однак PACF може вказувати на велике значення часткової автокореляції з лагом 17, але такий великий порядок для авторегресивної моделі, ймовірно, не має великого сенсу[5].

1.2 Регресія з авторегрессивними похибками

Далі, давайте розглянемо задачу, в якій ми маємо y - змінну та x - змінну, усі виміряні як часовий ряд. Як приклад, ми можемо мати y як місячні аварії на міждержавній автомагістралі, а x як місячну кількість проїзду між містом, з вимірюваннями, які спостерігаються протягом 120 місяців поспіль. Модель множинної (часових рядів) регресії можна записати як:

$$y_t = X_t\beta + \varepsilon_t \quad (1.4)$$

Труднощі, які часто виникають у цьому контексті, полягають в тому, що похибки (ε_t) можуть бути пов'язані одна з одною. Іншими словами, ми маємо автокореляцію або залежність між похибками[3, 4, 5].

Ми можемо розглянути ситуації, коли помилка в певний момент часу лінійно пов'язана з помилкою в попередній момент. Тобто самі помилки відповідають простій моделі лінійної регресії, яку можна записати як:

$$\varepsilon_t = \rho\varepsilon_{t-1} + \omega_t \quad (1.5)$$

Тут $|\rho| < 1$ називається параметром автокореляції, а ω_t — це новий термін похибки, який відповідає звичайним припущенням, які ми робимо щодо помилок регресії: $\omega_t \sim_{iid} N(0, \sigma^2)$. (Тут "iid" означає "незалежний і однаково розподілений.) Отже, ця модель говорить, що помилка в момент t є передбачуваною з частки помилки в момент часу $t-1$ плюс деяке додаткове збурення ω_t.

Наша модель для ε_t похибок вихідної регресії y, яка залежить від x, яка є авторегресивною моделлю похибок, зокрема AR(1) у цьому випадку. Однією з причин, чому похибки можуть мати авторегресивну структуру, є те, що змінні y і x в момент t можуть бути (і, швидше за все, є) пов'язані з вимірюваннями y і x в момент $t-1$. Ці співвідношення включаються і в член похибки нашої моделі множинної лінійної регресії. Зауважте, що авторегресивна модель помилок є порушенням припущення(Гаусса-Маркова), що ми маємо незалежні помилки, і це створює теоретичні труднощі для звичайних оцінок за методом найменших квадратів β-коефіцієнтів. Теорема Гаусса-Маркова стверджує, що у звичайному методі найменших квадратів оцінювач має найменшу дисперсію вибірки в межах класу від лінійних неупереджених оцінок, якщо похибки у лінійній регресійній моделі є некорильованими, мають рівні дисперсії та очікуване значення нуля. Існує кілька різних методів для оцінки параметрів регресії співвідношення y проти x, коли ми маємо помилки з авторегресивною структурою.

Похибки ε_t все ще мають середнє 0 і постійну дисперсію:

$$E(\varepsilon_t) = 0 \quad (1.6)$$

$$Var(\varepsilon_t) = \frac{\sigma^2}{1-\rho^2} \quad (1.7)$$

Однак коваріація (міра зв'язку між двома змінними) між сусідніми доданками помилки є:

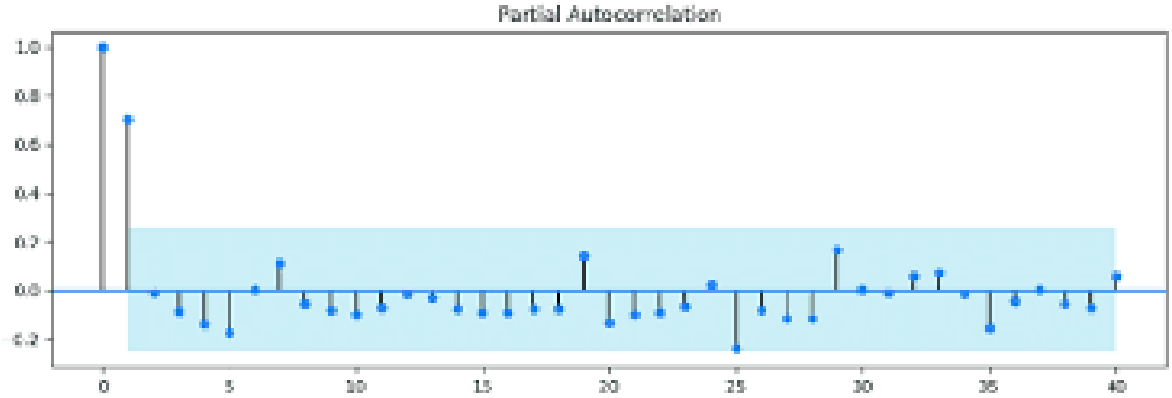
$$Cov(\varepsilon_t, \varepsilon_{t-1}) = \rho\left(\frac{\sigma^2}{1-\rho^2}\right) \quad (1.8)$$

що означає, що коефіцієнт кореляції (міра зв'язку між двома змінними) між сусідніми елементами похибки дорівнює:

$$Corr(\varepsilon_t, \varepsilon_{t-1}) = \frac{Cov(\varepsilon_t, \varepsilon_{t-1})}{\sqrt{Var(\varepsilon_t)Var(\varepsilon_{t-1})}} = \rho \quad (1.9)$$

який є параметром автокореляції, який ми ввели вище.

Ми можемо використовувати графіки функції часткової автокореляції (PACF), щоб допомогти нам оцінити відповідні лаги для помилок у моделі регресії з помилками авторегресії. Зокрема, ми спочатку нашу модель множинної лінійної регресії будуємо відповідно до даних часових рядів і зберігаємо залишки. Потім ми можемо поглянути на графік PACF для залишків у порівнянні з лагом. Велика вибірка часткових автокореляцій, які значно відрізняються від 0, вказують на відставання термінів ε, які можуть бути корисними предикторами ε_t.



Мал. 2.1. Приклад графіку PACF

1.3 Експоненціальне згладжування

Експоненціальне згладжування було запропоновано наприкінці 1950-х років (Браун, 1959; Холт, 1957; Вінтерс, 1960) і спонукало до деяких найуспішніших методів прогнозування. Прогнози, створені за допомогою методів експоненціального згладжування, є середніми зваженими показниками минулих спостережень, причому вагові коефіцієнти експоненціально зменшуються в міру старіння спостережень. Іншими словами, чим новіше було спостереження, тим вище пов’язана вага. Ця структура швидко генерує надійні прогнози та для широкого діапазону часових рядів, що є великою перевагою та має велике значення для застосування в промисловості.

Експоненціальне згладжування, ймовірно, є широко використовуваним класом процедур для згладжування дискретних часових рядів для прогнозування найближчого майбутнього. Цю популярність можна

пояснити його простотою, обчислювальною ефективністю, простотою регулювання його чутливості до змін прогнозованого процесу та розумною точністю[8].

Ідея експоненціаль ного згладжування полягає в тому, щоб згладити вихідний ряд так, як це робить ковзне середнє, і використовувати згладжений ряд для прогнозування майбутніх значень змінної, що цікавить. Однак у експоненціальному згладжуванні ми хочемо дозволити новіші значення ряду мати більший вплив на прогноз майбутніх значень, ніж більш віддалені спостереження[9].

Цей метод прогнозування найбільш широко використовується з усіх методів прогнозування. Для цього потрібно мало обчислень. Цей метод використовується, коли шаблон даних є приблизно горизонтальним (тобто в історичних даних немає ні циклічних змін, ні яскраво вираженої тенденції).

Нехай спостережуваний часовий ряд буде $y_1, y_2, \dots, y_n$. Формально просте рівняння експоненціального згладжування набуває вигляду

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t \quad (1.10)$$

де y_t – фактичне, відоме значення ряду для періоду часу t, $\hat{y}_t$ – прогнозне значення змінної Y для періоду часу t, $\hat{y}_{t+1}$ – прогнозне значення для періоду часу $t+1$, а α – значення константа згладжування. Прогноз $\hat{y}_{t+1}$ заснований на зважуванні останнього спостереження y_t з вагою α і зважування останнього прогнозу $\hat{y}_{t+1}$ з вагою $1-\alpha$.

Щоб почати роботу з алгоритмом, нам потрібен початковий прогноз, фактичне значення і константа згладжування. Оскільки $\hat{y}_1$ невідоме, ми можемо:

1. Встановити першу оцінку рівною першому спостереженню. Таким чином, ми можемо використовувати $\hat{y}_1 = y_1$.

2. Використати середнє з перших п’яти або шести спостережень для початкового згладженого значення.

Константа згладжування α — це вибране число від нуля до одиниці, $0 < \alpha < 1$. Коли $\alpha = 1$, вихідна та згладжена версія ряду ідентичні. З іншого боку, коли $\alpha = 0$, ряд згладжується плоскою.

Перепишемо модель, щоб побачити одну з чудових речей моделі SES

$$\hat{y}_{t+1} - \hat{y}_t = \alpha(y_t - \hat{y}_t) \quad (1.11)$$

зміна прогнозного значення пропорційна помилці прогнозу. Це

$$\hat{y}_{t+1} = \hat{y}_t + \alpha\varepsilon_t \quad (1.12)$$

де залишок $\varepsilon_t = y_t - \hat{y}_t$ – є помилкою прогнозу для періоду часу t. Отже, прогноз експоненціального згладжування – це старий прогноз плюс коригування на помилку, що сталася в останньому прогнозі[9].

Продовжуючи замінювати попередні значення прогнозу назад до початкової точки даних у моделі (1), ми отримуємо:

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)[\alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1}] = \alpha y_t + \alpha(1-\alpha)y_{t-1} + (1-\alpha)^2\hat{y}_{t-1},$$

$$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + (1-\alpha)^2 y_{t-2} + (1-\alpha)^3\hat{y}_{t-2},$$

Рівняння прогнозу в загальному вигляді

$$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + (1-\alpha)^2 y_{t-2} + \cdots + (1-\alpha)^{t-2}y_2 + (1-\alpha)^{t-1}y_1$$

$$= \alpha\sum_{k=0}^{t-1}(1-\alpha)y_{t-k} \quad (1.13)$$

де $\hat{y}_{t+1}$ – це прогнозне значення змінної Y на період часу $t + 1$ на основі знання фактичних значень ряду y_t, y_{t-1}, y_{t-2} і так далі назад у часі до першого відомого значення часового ряду y_1. Отже, $\hat{y}_{t+1}$ є зваженим ковзним середнім усіх минулих спостережень.

Ряд вагових коефіцієнтів, що використовуються для створення прогнозу $\hat{y}_{t+1}$, дорівнює $\alpha, \alpha(1-\alpha), \alpha(1-\alpha)^2, \ldots$. Ці ваги зменшуються до нуля експоненціально; таким чином, коли ми повертаємося до серії, кожне значення має меншу вагу з точки зору його впливу на прогноз. Експоненціальне зниження ваг до нуля є очевидним.

Подвійне експоненційне згладжування - цей метод також відомий як метод Холта на честь Чарльза К. Холта та його статті 1957 року.

Це називається подвійним експоненціальним згладжуванням, оскільки воно засноване на двох параметрах згладжування — α (для рівня) і β (для тренду). Алгоритм вирішує основну проблему простого експоненційного згладжування, оскільки тепер прогнози можуть враховувати тенденцію в історичних даних.

Говорячи про тенденцію, вона може бути як адитивною, так і мультиплікативною. Адитивна тенденція — тенденція лінійно зростає з часом. Мультиплікативна тенденція — тенденція не росте лінійно і демонструє кривизну — навіть незначну.

Математично подвійне експоненціальне згладжування можна виразити такою формулою:

$$\text{Рівень: } l_t = (1-\alpha)l_{t-1} + \alpha y_t$$
$$\text{Тренд: } b_t = (1-\beta)b_{t-1} + \beta(l_t - l_{t-1})$$
$$\text{Прогнозування: } \hat{y}_{t+n} = l_t + nb_t \quad (1.14)$$

де n представляє кількість кроків у часі в майбутнє. α і β є параметрами згладжування.

Однак подвійне експоненціальне згладжування не може вирішити один важливий компонент набору даних - сезонність. Ось тут і з'являється потрійне експоненціальне згладжування.

Через три роки (1960) Пітер Р. Вінтерс і Чарльз. К. Холт розширили оригінальний метод Холта, щоб урахувати сезонність. Алгоритм був названий на честь обох — метод Холта-Вінтерса.

Ще один параметр був доданий - γ - для адресації сезонної складової. Алгоритм також вимагає вказати кількість періодів в одному сезонному циклі. У більшості ресурсів він позначається буквою L.

Так само, як і тенденція, сезонність також може бути адитивною або мультиплікативною. Математично потрійне експоненціальне згладжування можна виразити такою формулою:

$$\text{Рівень: } l_t = (1-\alpha)l_{t-1} + \alpha y_t$$
$$\text{Тренд: } b_t = (1-\beta)b_{t-1} + \beta(l_t - l_{t-1})$$
$$\text{Сезонність: } c_t = (1-\gamma)c_{t-L} + \gamma(y_t - l_{t-1} - b_{t-1})$$
$$\text{Прогнозування: } \hat{y}_{t+n} = (l_t + nb_t)c_{t-L+1+(n-1)\mathrm{mod}L} \quad (1.15)$$

Після визначення моделі її характеристики продуктивності мають бути перевірені або підтверджені шляхом порівняння її прогнозу з історичними даними для процесу, для якого вона була розроблена. Ми можемо використовувати такі показники помилки, як MAPE (середня абсолютна відсоткова помилка), MSE (середньоквадратична помилка) або RMSE (середньоквадратична помилка):

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{y_t}\cdot 100\% \quad (1.16)$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \cdot 100\% \quad (1.17)$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \quad (1.18)$$

Вибір міри помилки має важливий вплив на висновки про те, який із набору методів прогнозування є найбільш точним.

Швидкість, з якою старіші реакції згладжуються, є функцією значення α. Коли константа згладжування α близька до 1, демпфування відбувається швидко, а коли α близьке до 0, демпфування повільне[9]. Якщо ми хочемо, щоб прогнози були стабільними, а випадкові варіації згладжувалися, використовуйте мале α. Якщо ми хочемо швидкої реакції, потрібно більше значення α.

Зазвичай MSE або RMSE можна використовувати як критерій для вибору відповідної константи згладжування. Наприклад, призначаючи значення α від 0,1 до 0,99, ми вибираємо значення, яке дає найменшу MSE або $RMSE$[8].

1.4 Прямі та ітеровані багатокрокові методи AR для прогнозування

У цьому розділі описуються ітераційні та прямі моделі та оцінки прогнозування.

«Ітераційні» багатоперіодні прогнози наперед складаються з використанням моделі на один період, що повторюється вперед протягом бажаної кількості періодів, тоді як «прямі» прогнози робляться з використанням моделі оцінювання для конкретного горизонту, де залежною змінною є багаторазово прогнозоване значення на період вперед. (вставити аналітичний приклад) Який підхід є кращим, є емпіричним питанням: теоретично ітераційні прогнози є ефективнішими, якщо їх правильно вказати, але прямі прогнози більш надійні щодо неправильної специфікації моделі. Ітераційні прогнози зазвичай перевершують прямі прогнози, особливо якщо моделі можуть вибрати специфікації з довгим лагом. Відносна продуктивність ітераційних прогнозів покращується разом із горизонтом прогнозу[8].

Почнемо з двох загальних спостережень.

По-перше, багато рядів часу видаються нестаціонарними в тому сенсі, що мають один або кілька одиничних коренів. Стратегія, прийнята тут, полягає в перетворенні ряду, що цікавить, до наближеної стаціонарності, взявши його першу або другу різницю за потребою, щоб оцінити модель прогнозування, а

потім обчислити прогноз на крок вперед початкового ряду, створеного цією моделлю. Наприклад, логарифм реального ВВП спочатку перетворюється, взявши його першу різницю, $\Delta\log\text{ВВП}_t$, прогнозна модель оцінюються за допомогою $\Delta\log\text{ВВП}_t$, а потім ці моделі використовуються для обчислення прогнозу рівня логарифму ВВП, h періодів. попереду. Перетворення, які використовуються для кожної серії, обговорюватимуться далі.

По-друге, усі прогнози є рекурсивними (псевдо - за межами вибірки), тобто прогнози ґрунтуються лише на значеннях ряду до дати, на яку зроблено прогноз.

Потім параметри переоцінюються в кожному періоді для кожної моделі прогнозування, використовуючи дані від початку вибірки до поточної дати прогнозування. Для прогнозів, що передбачають вибір моделі на основі даних, порядок моделі також вибирається рекурсивно, і, таким чином, може змінюватися у вибірці, коли до набору даних прогнозу додається нова інформація[10].

Нехай X_t позначає рівень або логарифм ряду, що цікавить. Метою є обчислення прогнозів X_{t+h}, використовуючи інформацію в момент часу t. Нехай y_t позначає стаціонарне перетворення ряду після взяття першої або другої різниць. Зокрема, припустимо, що X_t інтегрується від порядку d (є $I(d)$); тоді $y_t = \Delta^d X_t$, де $d = 0, 1$ або 2 відповідно.

Модель AR на крок вперед для y_t є

$$y_{t+1} = \alpha + \sum_{i=1}^{p} \varphi_i y_{t+1-i} + \varepsilon_t \quad (1.19)$$

Для ітераційних прогнозів AR параметри $\alpha, \varphi_1, \dots, \varphi_p$, в (1.19) оцінюються рекурсивно за OLS, а прогнози y_{t+h} будуються рекурсивно як,

$$\hat{y}^I{}_{t+h/t} = \hat{\alpha} + \sum_{i=1}^{p} \hat{\varphi}_i \hat{y}^I{}_{t+h+i/t} \quad (1.20)$$

де $\hat{y}_{j/t} = y_j$ для j ≤ t. Прогнози $\text{X}_{\text{t+h}}$ потім обчислюються шляхом накопичення значень $\hat{y}^I{}_{t+k/t}$ відповідно до випадків I(0), I(1) і I(2):

$$X_{t+h|t} = \begin{cases} \hat{y}^I{}_{t+h|t} \text{якщо } X_t \in I(0) \\ X_t + \sum_{i=1}^{h} \hat{y}^I{}_{t+i|t} \text{ якщо } X_t \in I(1) \\ X_t + \sum_{i=i}^{h} \sum_{j=1}^{i} \hat{y}^I{}_{t+j|t} \text{ якщо } X_t \in I(2) \end{cases} \quad (1.21)$$

Прямі оцінки параметрів є рекурсивними мінімізаторами середньоквадратичної помилки функції критерію h-кроку вперед.

Відповідно, параметри оцінюються за допомогою регресії OLS, в якій регресори є константою і $y_t, \dots, y_{t-p+1}$, а y_{t+h}^h залежна змінна, де

$$y_{t+h}^{h} = \begin{cases} X_{t+h} \text{якщо } X_t \in I(0) \\ X_{t+h} - X_t \text{якщо } X_t \in I(1) \\ \sum_{i=i}^{h} \sum_{j=1}^{i} \Delta^2 X_{t+j} = X_{t+h} - X_t - h\Delta X_t \text{якщо } X_t \in I(2) \end{cases} \quad (1.22)$$

Регресійна модель прямого прогнозування:

$$y_{t+h}^{h} = \beta + \sum_{i=1}^{p} \rho_i y_{t+1-i} + \varepsilon_{t+h} \quad (1.23)$$

Пряму оцінку коефіцієнтів отримують шляхом рекурсивної оцінки (1.21) за допомогою OLS, де використовуються дані за період t (так що останнє спостереження включає y_t^h з лівого боку регресії). Прямі прогнози y_{t+h}^h є

$$\widehat{y_{t+h}}^{D,h} = \hat{\beta} + \sum_{i=1}^{p} \hat{\rho}_i y_{t+1-i} \quad (1.24)$$

Прогнози X_{t+h} потім обчислюються з $\widehat{y_{t+h}}^{D,h}$, відповідно до випадків I(0), I(1) та I(2): $\widehat{X_{t+h/t}}^{D} = \widehat{y_{t+h/t}}^{D,h}$ для I(0), $\widehat{X_{t+k/t}}^{D} = \widehat{y_{t+h}}^{D,h} + X_t$ для I(1) і $\widehat{X_{t+h/t}}^{D} = \widehat{y_{t+h}}^{D,h} + X_t + h\Delta X_t$ для I(2).

Для визначення порядку лагу p використовують 4 різні методи: $p = 4$(фіксоване); $p = 12$ (фіксоване); p обраний інформаційним критерієм Акайке (AIC), з $0 \le p \le 12$, і p, вибраним інформаційним критерієм Байєса (BIC), з $0 \le p \le 12$.) Для ітерованих прогнозів AIC і BIC можна обчислювати за стандартними формулами, заснованими на сумі квадратів залишків (SSR) з регресії на один крок вперед. Для прямих прогнозів AIC і BIC можна розраховувати за допомогою SSR на основі оціненої регресії на крок вперед (1.23). Якщо AIC і BIC перераховувати на кожну дату, то порядок вибраної моделі прогнозування може змінюватися від одного періоду до наступного, де вибір моделі та оцінки параметрів базуються лише на даних до дати прогнозу (період t) [5].

$$AIC = -2\log(L) + 2k \quad (1.25)$$

де L є ймовірність моделі і k – загальна кількість оцінених параметрів і початкових станів.

$$BIC = \text{AIC} + \text{k}[\log(\text{T}) - 2] \quad (1.26)$$

Ці чотири варіанти охоплюють основні випадки, що становлять теоретичний інтерес. Якщо справжній порядок лагу скінченний і якщо максимальний розглянутий лаг перевищує p_0, то BIC забезпечує послідовну оцінку p_0, а повторна оцінка з BIC є асимптотично ефективною. Якщо p_0 є нескінченним, то пряма оцінка з вибором моделі AIC досягає межі ефективності для прямих оцінок, і ця межа нижче, ніж для всіх повторюваних оцінок. Однак у кінцевих вибірках вибір довжини затримки BIC і AIC вносить додаткову невизначеність вибірки, а короткі (4 лаги) і довгі (12 лаг) авторегресії з фіксованим лагом забезпечують контрольні показники, з якими можна порівнювати прогнози BIC і AIC.

Нехай t_0 позначає перше спостереження, використане для оцінки регресій, t_1 позначає дату, на яку робиться перший псевдопрогноз поза вибіркою, а t_2 позначає дату, коли робиться остаточний прогноз поза вибіркою. Дата t_0 — це дата, на яку доступне перше спостереження (для більшості серій), плюс дванадцять (оскільки для моделей із довгим лагом використовуються дванадцять лаг), а також порядок інтеграції серії (щоб дозволити для першої та другої відмінностей). Остаточна дата прогнозу залежить від горизонту прогнозу і є датою останнього доступного спостереження мінус горизонт прогнозу h.

Псевдопомилка прогнозу поза вибіркою становить $e_{t+h} = \hat{y}_{t+h} - y_{t+h}$, а вибіркова MSFE дорівнює,

$$MSFE = \frac{1}{t_2 - t_1 + 1} \sum_{i=t_1}^{t_2} e_{i+h}^2 \quad (1.27)$$

Вибірковий MSFE обчислюється для кожної серії, для кожного методу прогнозування. Для даної серії мпірична ефективність порівнянних прямих і непрямих прогнозів оцінюється шляхом порівняння відповідних MSFE. Вибірковий MSFE може бути меншим для прямого, ніж ітерованого прогнозу або тому, що прямий прогноз є більш ефективним у сукупності, або через мінливість вибірки[10].

РОЗДІЛ 2. Прогнозування

2.1 Прогнозування AR моделі для стаціонарних даних

З метою практичного вивчення прогнозування була поставлена задача на знаходження за допомогою прогнозування AR моделей, експоненціального згладжування та порівняння цих методів з реальними данными. Для досягнення задач будемо використовувати мову програмування Python.

Для прогнозу за допомогою AR моделі нам потрібно виконати декілька кроків:

1. Візуалізувати дані часових рядів
2. Визначити, чи є дані стаціонарні. Якщо ні, перетворити їх на стаціонарні.
3. Побудувати діаграми кореляції та автоматичної кореляції
4. Побудувати модель ARIMA або сезонну ARIMA на основі даних

В якості часового ряду, ми обрали ціну на золото[1]. Зобразимо графічно початковий ряд.

Мал. 2.1. Графік ціни на золото

Маємо зростаючий тренд до 2012 року, і спадаючий з 2013 по 2016.

Для ARIMA перше, що ми робимо, це визначаємо, є дані стаціонарними чи нестаціонарними. якщо дані нестаціонарні, ми спробуємо зробити їх стаціонарними, потім обробимо далі.

Щоб визначити природу даних, ми будемо використовувати нульову гіпотезу.

H_0: Нульова гіпотеза: це твердження про сукупність, яке або вважається істинним, або використовується для висунення аргументу, якщо не буде доведено, що воно є неправильним поза розумним сумнівом.

H_1: Альтернативна гіпотеза: це твердження про сукупність, яке суперечить H_0, і те, що ми робимо, коли відкидаємо H_0.

$\#H_0$: Він нестаціонарний

$\#H_1$: Він стаціонарний

Ми будемо розглядати нульову гіпотезу про те, що дані не є стаціонарними, і альтернативну гіпотезу про те, що дані є стаціонарними.

Проведемо кілька тестів для визначення стаціонарності.

1. **Тест Шапіро-Уілка**

   Тест Шапіро-Уілка кількісно визначає схожість даного розподілу і нормального лише одним числом. Далі крива нормального розподілу накладається на гістограму початкових даних - обраховується відсоток даних, що сходяться з нормальним розподілом, і, нарешті, імовірність сходження наших даних з нормальним розподілом. Тест є сильним, оскільки він використовується лише для порівняння з нормальним розподілом.

   Отримали такі результати:

   W-статистика = 0.9120041728019714
   p = 1.6617995779998334e-34

   Значення p значно менше за 5%, тому можемо відхилити нульову гіпотезу тесту.

2. **Тест Колгоморова-Смірнова**

   Тест Колмогорова-Смірнова обраховує відстані між емпіричним і теоретичним розподілами і визначає статистику тесту як верхню границю набору цих відстаней. Проте тест не є сильним, оскільки використовується не тільки для порівняння з нормальним розподілом, на відміну від теста Шапіро-Уілка.

   Отримали такі результати:

   D-статистика = 0.5556897585302352
   p = 0.0

Значення p = 0, тому можемо відхилити нульову гіпотезу тесту.
Обидва тести показали, що ми можемо відкинути гіпотезу про те, що наш часов ий ряд розподілений нормально.

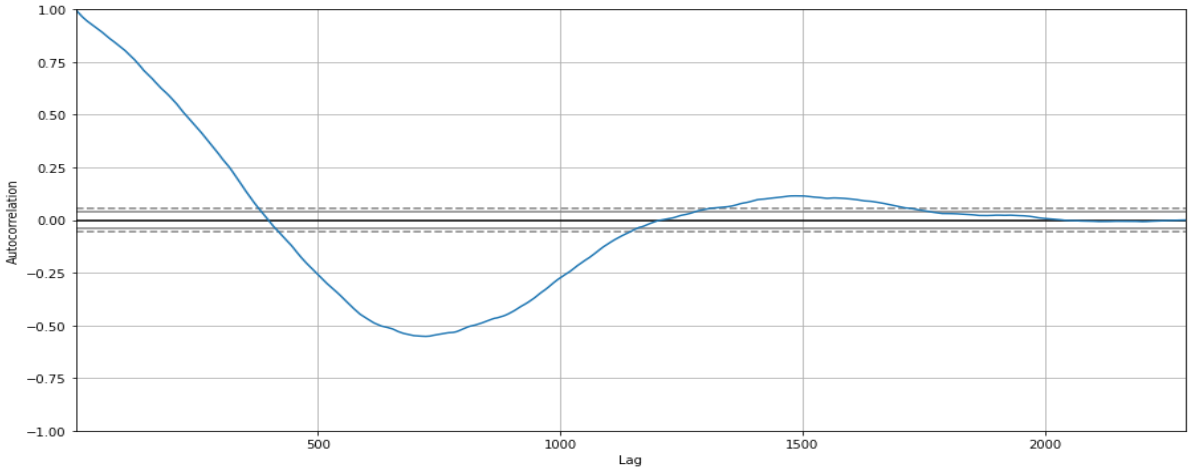
3. Тест Дікі-Фулера

Тест Дікі-Фулера припускає, що наш ряд описується лінійною функцією $y_t = M + \varphi_1 y_t + \varepsilon_t$, тобто що кожне наступне значення ряду залежить від попереднього. Нульова гіпотеза тесту: коефіцієнт $\varphi_1 \geq 1$, тобто ряд не є стаціонарним. Результат тесту - значення "Тест статистики" і кілька критичних значень тесту для деяких рівнів довіри. Якщо "Тест-статистика" менша за певне критичне значення, можемо стверджувати з відповідною імовірністю, що ряд є стаціонарним.

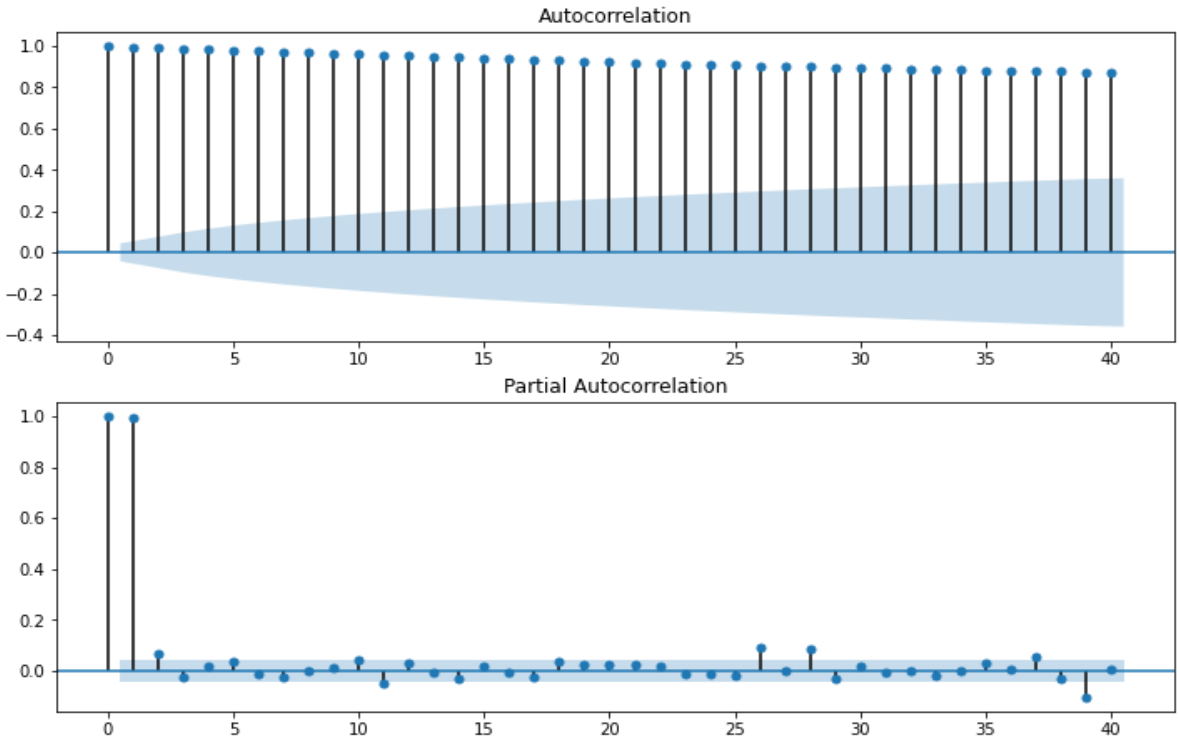
```
Results of the Dickey-Fuller Test:
Test Statistic           -1.094117e+01
p-value                   9.288085e-20
# of Lags Used            2.600000e+01
# of Observations Used    2.262000e+03
Critical Value (1%)      -3.433244e+00
Critical Value (5%)      -2. 862819e+00
Critical Value (10%)     -2.567451e+00
      dtype: float64
```

Значення p значно менше за 5%, тож можемо відхилити нульову гіпотезу тесту і сказати, що ряд схожий на стаціонарний.

Створимо автокореляцію:



Мал. 2.2. Автокорреляція



Мал. 2.3. Графіки ACF та PACF ціни на золото

Підберемо значення p, q, d, P, D, Q для нашої моделі за допомогою AIC та BIC, отримаємо таку табличку:

Таблиця 2.1 Значення AIC для різних ARIMA моделей

ARIMA	*AIC*
(1,0,1)(0,1,1)[12]	inf
(0,0,0)(0,1,0)[12]	13897.428
(1,0,0)(1,1,0)[12]	9282.096
(0,0,1)(0,1,1)[12]	11874.672
(0,0,0)(0,1,0)[12]	13898.850
(1,0,0)(0,1,0)[12]	9886.825
(1,0,0)(2,1,0)[12]	9048.693
(1,0,0)(2,1,1)[12]	inf
(1,0,0)(1,1,1)[12]	inf

(0,0,0)(2,1,0)[12]	13858.992
(2,0,0)(2,1,0)[12]	9050.233
(1,0,1)(2,1,0)[12]	9050.263
(0,0,1)(2,1,0)[12]	11869.192
(2,0,1)(2,1,0)[12]	9051.346
(1,0,0)(2,1,0)[12]	9046.942
(1,0,0)(1,1,0)[12]	9280.290
(0,0,0)(2,1,0)[12]	13861.966
(2,0,0)(2,1,0)[12]	9048.475
(1,0,1)(2,1,0)[12]	9048.505
(0,0,1)(2,1,0)[12]	11871.715
(2,0,1)(2,1,0)[12]	9049.588

Отримали $ARIMA(1,0,0)(2,1,0)_{12}$. Побудуємо нашу модель.

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9536	0.005	208.431	0.000	0.945	0.963
ar.S.L12	-0.6460	0.013	-49.764	0.000	-0.671	-0.621
ar.S.L24	-0.3153	0.014	-22.262	0.000	-0.343	-0.288
sigma2	3.0857	0.039	79.907	0.000	3.010	3.161

Ljung-Box (L1) (Q):	0.41	**Jarque-Bera (JB):**	8591.19
Prob(Q):	0.52	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.48	**Skew:**	-0.80
Prob(H) (two-sided):	0.00	**Kurtosis:**	12.38

Мал. 2.4. Модель ARIMA для ціни на золото

Наша модель:

$$y_t = 3.0857 + 0.9536y_{t-1} - 0.646y_{t-12} - 0.3153y_{t-24} + \varepsilon_t$$

Спрогнозуємо останні 10 значень, отримаємо:

```
Date
4/30/2018    125.365659
5/1/2018     126.131202
5/2/2018     126.749738
5/3/2018     126.513095
5/7/2018     127.004574
5/8/2018     126.830348
5/9/2018     126.290167
5/10/2018    126.303232
5/14/2018    126.068854
5/16/2018    125.720637
Name: forecast, dtype: float64
```

Мал. 2.5. Спрогнозовані значення ціни на золото

2.2 Прогнозування AR моделі для нестаціонарних даних

Виконаємо ті самі кроки, що і для стаціонарних даних. В якості датасету було обрано щотижневі ціни на акції компанії Samsung[2]. Зобразимо графічно початковий ряд.

Мал. 2.6. Графік цін акцій Samsung

Одразу бачимо зростаючий тренд. Проведемо тест для визначення стаціонарності.

Тест Дікі-Фулера:

```
ADF Test Statistic : -0.8262855381215608
p-value : 0.8112787036664966
#Lags Used : 18
Number of Observations : 971
weak evidence against null hypothesis,indicating it is non-stationary
```

Мал. 2.7. Результати тесту Дікі-Фулеру для цін Samsung

Значення p значно більше за 5%, тож ми не можемо відхилити нульову гіпотезу тесту і сказати, що ряд стаціонарний.

Давайте спробуємо взяти першу різницю та проведемо тест.

Тест Дікі-Фулера:

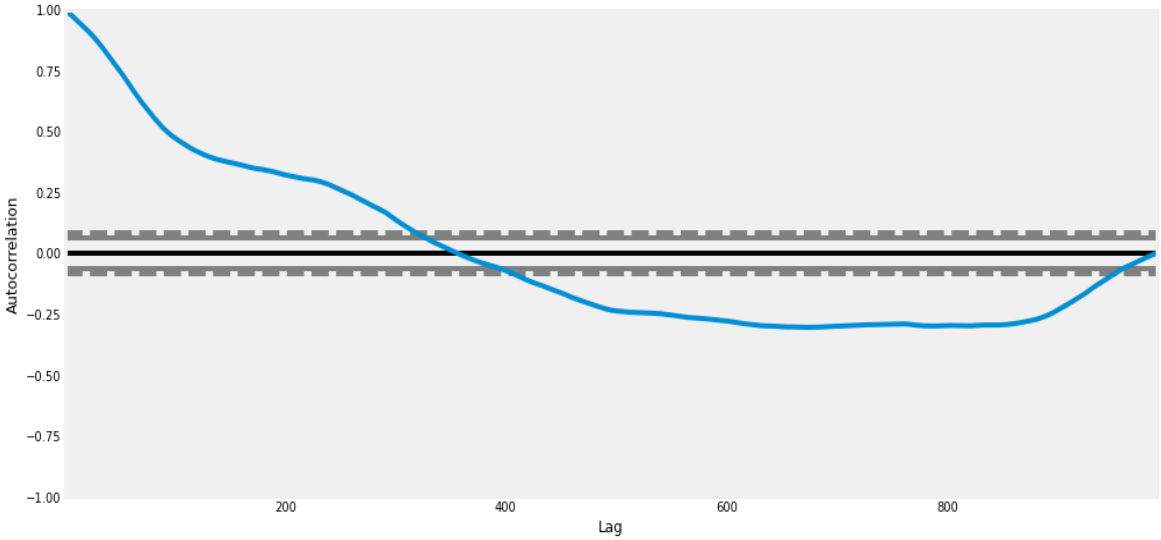
```
ADF Test Statistic : -4.974016486993952
p-value : 2.4976651665518988e-05
#Lags Used : 20
Number of Observations : 957
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is stationary
```

Мал. 2.8. Результати тесту Дікі-Фулеру для першої різниці

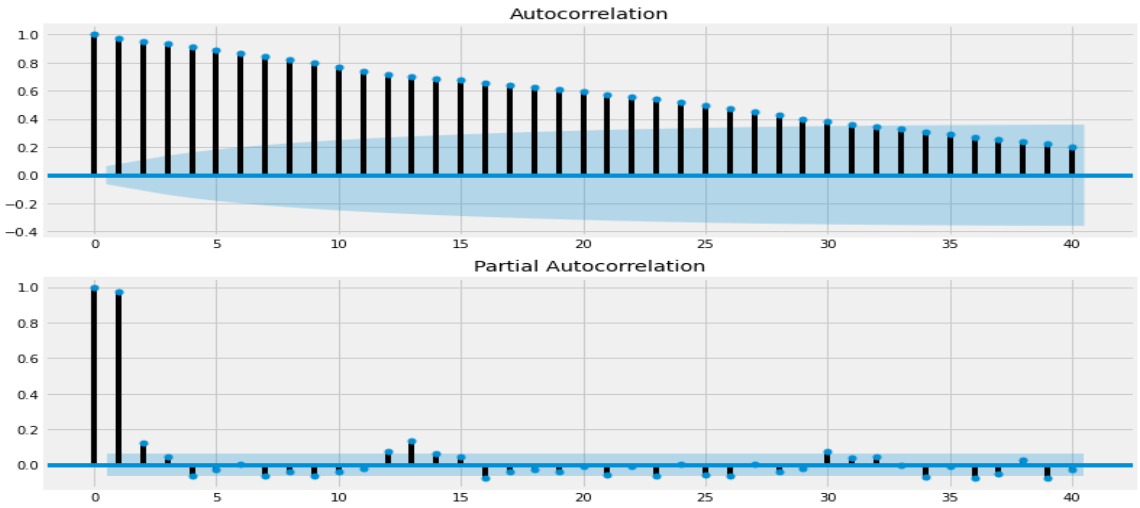
Значення p значно менше за 5%, що означає, що ми можемо відхилити нульову гіпотезу. Отже, дані стаціонарні.

Мал. 2.9. Графік першої різниці цін акцій Samsung

Створимо автокореляцію:



Мал. 2.10. Автокорреляція



Мал. 2.11. Графіки ACF та PACF

Підберемо значення p, d, q, P, D, Q для нашої моделі за допомогою AIC, отримаємо таку табличку:

Таблиця 2.2. Значення AIC для різних ARIMA моделей

ARIMA	*AIC*
(3,0,3)(0,1,1)[12]	inf
(0,0,0)(0,1,0)[12]	18115.916
(1,0,0)(1,1,0)[12]	16329.220
(0,0,1)(0,1,1)[12]	17325.803

(0,0,0)(0,1,0)[12]	18142.242
(1,0,0)(0,1,0)[12]	16653.976
(1,0,0)(2,1,0)[12]	16253.527
(1,0,0)(2,1,1)[12]	16068.064
(0,1,0)(0,1,0)[12]	16673.058
(1,1,0)(1,1,0)[12]	16305.488
(1,1,0)(0,1,0)[12]	16665.884
(0,0,0)(2,1,1)[12]	18101.608
(0,1,0)(2,1,0)[12]	16235.069
(2,1,0)(1,1,0)[12]	16291.854
(0,0,1)(2,1,1)[12]	17321.776
(3,1,0)(2,1,0)[12]	16198.000
(1,0,0)(2,1,1)[12]	inf

Отримали $ARIMA(1,0,0)(2,1,1)_{12}$. Побудуємо нашу модель.

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.0000   4.29e-05   2.33e+04      0.000       1.000       1.000
ar.S.L12      -0.0733      0.022     -3.283      0.001      -0.117      -0.030
ar.S.L24       0.0406      0.021      1.926      0.054      -0.001       0.082
ma.S.L12      -0.9995      0.027    -36.718      0.000      -1.053      -0.946
sigma2      7.186e+05    3.9e-08   1.84e+13      0.000    7.19e+05    7.19e+05
===================================================================================
Ljung-Box (Q):                      121.26   Jarque-Bera (JB):               704.97
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               6.88   Skew:                             0.04
Prob(H) (two-sided):                  0.00   Kurtosis:                         7.16
===================================================================================
```

Мал. 2.12. Модель ARIMA для Samsung

Наша модель:

$$y_t = 718600 + y_{t-1} - 0.0733y_{t-12} - 0.0406y_{t-24} - 0.9995\varepsilon_{t-12} + \varepsilon_t$$

Спрогнозуємо останні 10 значень, отримаємо:

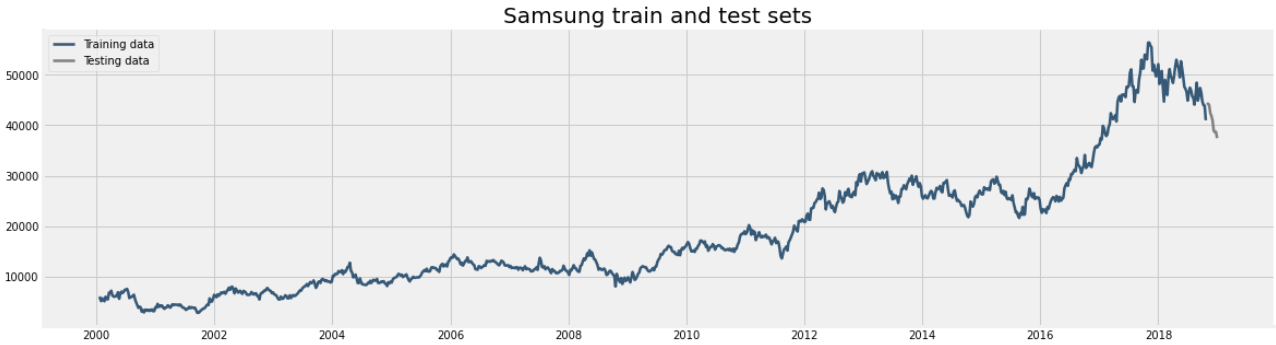
```
2018-10-29    40874.705968
2018-11-05    41166.474091
2018-11-12    41068.916652
2018-11-19    40884.481828
2018-11-26    41122.999597
2018-12-03    41172.445235
2018-12-10    40980.407839
2018-12-17    41036.374689
2018-12-24    41314.708360
2018-12-31    41369.047592
```

Мал. 2.13. Спрогнозовані значення для Samsung

Обраховане $RMSE = 4516.21, MAPE = 14.81\%$

2.3 Прогнозування методом TES

Завантажимо дані та поділимо наш часовий ряд на тренувальний та тестовий.



Мал. 2.14. Графічне представлення тренувальних та тестових даних

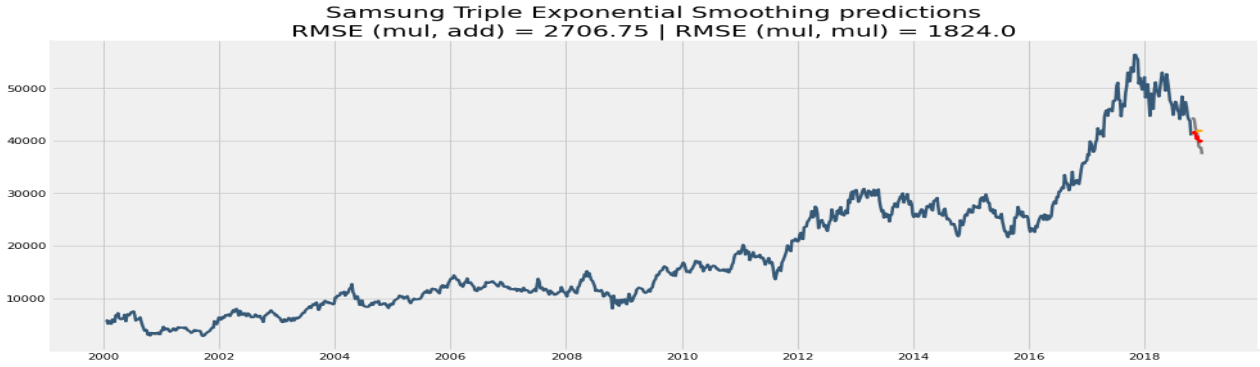
Побудуємо модель та спрогнозуємо значення.

$$\text{Рівень: } l_t = 0.37 l_{t-1} + 0.63 y_t$$

$$\text{Тренд: } b_t = 0.95 b_{t-1} + 0.05(l_t - l_{t-1})$$

$$\text{Сезонність: } c_t = 0.89 c_{t-L} + 0.11(y_t - l_{t-1} - b_{t-1})$$

$$\text{Модель: } y_t = (l_t + n b_t) c_t, \qquad l_0 = 18351.71, b_0 = 1.008, L = 12$$



Мал. 2.15. Графік прогнозування методом TES

Спрогнозовані значення:

2018-10-29	41516.097138
2018-11-05	41479.876547
2018-11-12	41715.399079
2018-11-19	41304.473124
2018-11-26	40340.212606
2018-12-03	40795.146650
2018-12-10	39991.422309
2018-12-17	39751.484077
2018-12-24	40024.273993
2018-12-31	39817.448222

Мал. 2.16. Прогнозовані значення методом TES

Обраховане $\mathrm{RMSE} = 1824, \mathrm{MAPE} = 5.92\%$

Висновки

Отже, в результаті проведеної роботи, в якій завдяки вивченні теорії та її аплікування до практики, досліджено різноманітну літературу стосовно часових рядів, методів їх опрацювання та отримання прогнозів. Досліджено вже відомі методи для знаходження рішення за різних умов.

У процесі роботи над кваліфікайціною роботою було розглянуто термінологію та основні положення необхідні для подальшого вивчення прогнозування часових рядів. Особливу увагу в роботі було приділено AR моделям, а саме ARIMA, та експоненціальному згладжуванню.

Використавши ці 2 методи на реальних даних, ми можемо зробити висновки на основі обчислених MSE та MAPE, що метод експонеціального згладжування продемонстрував себе майже вдвічі краще. З недоліків ARIMA моделей можна визначити:

1. Немає автоматичного оновлення. На відміну від простих наївних моделей або моделей згладжування, тут немає функції автоматичного оновлення. У міру появи нових даних всю процедуру моделювання необхідно повторити особливо на етапі діагностичної перевірки, оскільки модель могла зламатися.
2. Нестабільність. Модель ARIMA, як правило, нестабільна, як щодо змін у спостереженнях, так і щодо змін у специфікації моделі.
3. Прогнози працюють на короткий термін.

Переваги ARIMA:

1. Для прогнозу потрібні лише попередні дані часового ряду.
2. Добре працює з короткостроковими прогнозами.

Недоліки експоненціального згладжування:

1. Його прогноз буде відставати, оскільки тенденція з часом збільшується або зменшується.

2. Він не враховує динамічні зміни, які відбуваються на практиці. Його прогнози вимагатимуть постійного оновлення, щоб реагувати на нову інформацію.
3. Прогноз кожного місяця можна визначити лише після початку місяця, оскільки лише тоді стануть доступними фактичні дані за попередній місяць.
4. Експоненціальне згладжування неправильно обробляє тенденції. Цей метод найкраще підходить для короткострокового прогнозування, наприклад, наступного періоду. Він часто передбачає майбутні моделі, щоб значною мірою представляти поточні, тому не настільки ефективний у довгостроковому прогнозуванні.

Переваги:

1. Легко застосувати на практиці. Єдине, що потрібно для цього методу, — це прогноз на останній період часу, фактичні дані за цей період і константа згладжування.
2. Чим новіші дані, тим більше вони зважені в методі експоненційного згладжування. Це також означає, що стрибки в даних не так сильно впливають на прогноз.
3. Він дає точні прогнози. Метод експоненційного згладжування створює прогноз на один період вп–еред. Використовуючи техніку прогнозування тенденції, можна генерувати прогнози на більші періоди вперед. Прогноз вважається точним, оскільки враховує різницю між фактичними прогнозами та тим, що насправді відбулося.

Був розроблений алгоритм для реалізації прогнозування запрограмований на мові Python

Список використаної літератури

[1] Gold data set https://www.kaggle.com/altruistdelhite04/gold-price-data

[2] Samsung dataset until 2020. https://finance.yahoo.com/quote/005930.KS/history?period1=948240000&period2=1655596800&interval=1wk&filter=history&frequency=1wk&includeAdjustedClose=true

[3] Эрнст Роберт Берндт, Практика эконометрики. Класика та сучасність

[4] Лук'яненко, Економетрика

[5] Rob J Hyndman. Forecasting: Principles and Practice

[6] Чучуева И.А., “Модель прогнозирования временных рядов по выборке максимального подобия,” Москва, 2012. [Online].

[7] “5 Statistical Methods For Forecasting Quantitative Time Series | Bista INC,”

[8] BROWN, R. G., MEYER, R. F., The fundamental theory of exponential smoothing, Operations Research, 9, 1961, p. 673-685,

[9] MONTGOMERY, D. C., JOHNSON, L. A., GARDINER, J. S., Forecasting and Time Series Analysis, McGraw-Hill, Inc., 1990, ISBN 0-07-042858-1

[10] Bhansali, R.J. (1997, “Direct Autoregressive Predictions for Multistep Prediction: Order Selection and Performance Relative to the Plug In Predictors,”

Додаток до 2.1

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import matplotlib
import statsmodels.api as sm
import scipy.stats as stats
from sklearn.metrics import mean_squared_error
from matplotlib import rcParams
from cycler import cycler
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.gofplots import qqplot
plt.style.use('fivethirtyeight')
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.stattools import pacf
from scipy import fftpack
from numpy import fft
from statsmodels.tsa.seasonal import seasonal_decompose
from datetime import datetime
%matplotlib inline
df=pd.read_csv('gld.csv')
df.head()

# Updating the header
df.columns=["Date","GLD"]
df.head()
df.describe()
df.set_index('Date',inplace=True)

from pylab import rcParams
rcParams['figure.figsize'] = 15, 7
df.plot()

from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['GLD'].dropna(),lags=40,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['GLD'].dropna(),lags=40,ax=ax2)

import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['GLD'],order=(1, 0, 0),seasonal_order=(2,
1,0,12))
```

```
results=model.fit()
df['forecast']=results.predict(start=2280,end=2289,dynamic=True)
df[['GLD','forecast']].plot(figsize=(12,8))

print(df[-10:]['forecast'])
print(df[-10:]['GLD'])
```

Додаток до 2.2

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import matplotlib
import statsmodels.api as sm
import scipy.stats as stats
from sklearn.metrics import mean_squared_error
from matplotlib import rcParams
from cycler import cycler
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.gofplots import qqplot
plt.style.use('fivethirtyeight')
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.stattools import pacf
from scipy import fftpack
from numpy import fft
from statsmodels.tsa.seasonal import seasonal_decompose
from datetime import datetime
%matplotlib inline

df=pd.read_csv('sam.csv')
df.head()

# Updating the header
df.columns=["Date","Close"]
df.head()
df.describe()
df.set_index('Date',inplace=True)

from pylab import rcParams
rcParams['figure.figsize'] = 15, 7
df.plot()

from statsmodels.tsa.stattools import adfuller
test_result=adfuller(df['Close'])

def adfuller_test(sales):
    result=adfuller(sales)
    labels = ['ADF Test Statistic','p-
value','#Lags Used','Number of Observations']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )
    if result[1] <= 0.05:
```

```
        print("strong evidence against the null hypothesis(Ho), reject the n
ull hypothesis. Data is stationary")
    else:
        print("weak evidence against null hypothesis,indicating it is non-
stationary ")

adfuller_test(df['Close'])

df['Sales First Difference'] = df['Close'] - df['Close'].shift(1)
df['Seasonal First Difference']=df['Close']-df['Close'].shift(12)
df.head()
adfuller_test(df['Seasonal First Difference'].dropna())

df['Seasonal First Difference'].plot()

from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df['Close'])
plt.show()

from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['Seasonal First Difference'].dropna(),lags
=40,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Seasonal First Difference'].dropna(),lag
s=40,ax=ax2)

import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['Close'],order=(1, 0, 0),seasonal_order=(
2,1,1,12))
results=model.fit()

print(results.predict(start=980,end=989,dynamic=True))
print(df[-10:]['Close'])
rmse_mul = mean_squared_error(df[-
10:]['Close'], results.predict(start=980,end=989,dynamic=True), squared=Fals
e)
print('RMSE = ' + rmse)
print('MAPE = ' + MAPE(df[-
10:]['Close'], results.predict(start=980,end=989,dynamic=True)))
```

Додаток до 2.3

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import matplotlib
import statsmodels.api as sm
import scipy.stats as stats
from sklearn.metrics import mean_squared_error
from matplotlib import rcParams
from cycler import cycler
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.gofplots import qqplot
plt.style.use('fivethirtyeight')
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.stattools import pacf
from scipy import fftpack
from numpy import fft
from statsmodels.tsa.seasonal import seasonal_decompose
from datetime import datetime
%matplotlib inline


t = 10

rcParams['figure.figsize'] = 18, 5
rcParams['axes.spines.top'] = False
rcParams['axes.spines.right'] = False
rcParams['axes.prop_cycle'] = cycler(color=['#365977'])
rcParams['lines.linewidth'] = 2.5

# Load
df = pd.read_csv('sam.csv', index_col='Date', parse_dates=True)
df.index.freq = 'W-MON'

# Plot
plt.title('Samsung dataset', size=20)
plt.plot(df);


# Train/test split
df_train = df[:-t]
df_test = df[-t:]

# Plot
plt.title('Samsung train and test sets', size=20)
```

```
plt.plot(df_train['Close'], label='Training data')
plt.plot(df_test['Close'], color='gray', label='Testing data')
plt.legend();


# Multiplicative trend / Additive seasonality model
model_mul_add = ExponentialSmoothing(df_train['Close'], trend='mul', seasona
l='add', seasonal_periods=12)
results_mul_add = model_mul_add.fit()
predictions_mul_add = results_mul_add.forecast(steps=t)

# Multiplicative trend / Multiplicative seasonality model
model_mul_mul = ExponentialSmoothing(df_train['Close'], trend='mul', seasona
l='mul', seasonal_periods=12)
results_mul_mul = model_mul_mul.fit()
predictions_mul_mul = results_mul_mul.forecast(steps=t)

print(predictions_mul_mul)

# Evaluate
rmse_mul_add = mean_squared_error(df_test['Close'], predictions_mul_add, squ
ared=False)
rmse_mul_mul = mean_squared_error(df_test['Close'], predictions_mul_mul, squ
ared=False)
def MAPE(Y_actual,Y_Predicted):
    mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
    return mape

print(MAPE(df_test['Close'], predictions_mul_add))

# Plot
plt.title(f'Samsung Triple Exponential Smoothing predictions\nRMSE (mul, add
) = {np.round(rmse_mul_add, 2)} | RMSE (mul, mul) = {np.round(rmse_mul_mul,
2)}', size=20)
plt.plot(df_train['Close'], label='Training data')
plt.plot(df_test['Close'], color='gray', label='Testing data')
plt.plot(predictions_mul_add, color='orange', label='Predictions Multiplicat
ive trend, Additive seasonality')
plt.plot(predictions_mul_mul, color='red', label='Predictions Multiplicative
 trend, Multiplicative seasonality')
```