НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ

Кафедра математичних методів системного аналізу

До захисту допущено:

Завідувач кафедри
_____ Оксана ТИМОЩУК

«___»_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 «Системний аналіз»
на тему: «Прогнозування фінансових часових рядів. Порівняльний аналіз методів прогнозування»

Виконав:
студент IV курсу, групи КА-02
Макітрук Максим Тарасович _____

Керівник:
старший викладач, к.т.н.,Селін Юрій Миколайович _____

Консультант з економічного розділу:
професор, д.е.н, Семенченко Наталія Віталіївна _____

Консультант з нормоконтролю:
к.ф.-м.н., Статкевич Віталій Михайлович _____

Рецензент:
професор, д.т.н., Корнієнко Богдан Ярославович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Оксана ТИМОЩУК

«___»_____20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Макітруку Максиму Тарасовичу

1. Тема роботи «Прогнозування фінансових часових рядів. Порівняльний аналіз методів прогнозування», керівник роботи Селін Юрій Миколайович, кандидат технічних наук, старший викладач, затверджені наказом по університету від «___»_____ 2024 р. №_____

2. Термін подання студентом роботи 12.06.2024

3. Вихідні дані до роботи: курс золота та рух індексу S&P 500

4. Зміст роботи дослідження предметної області, опис методів і метрик для прогнозування фінансових часових рядів, приклад виконання прогнозування з залученням програмного продукту, функціонально-вартісний аналіз.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): графіки результатів прогнозування, інформаційні графіки даних, ілюстрації якості результатів прогнозування, презентація захисту.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Функціонально-вартісний аналіз	Семенченко Н. В., д.е.н, професор		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми ДР	21.04.2024	виконано
2	Збір та аналіз літератури	21.04.2024	виконано
3	Дослідження предметної області	28.04.2024	виконано
4	Аналіз актуальності дослідження	28.04.2024	виконано
5	Формулювання завдання дослідження	28.04.2024	виконано
6	Розробка програмного продукту та його тестування	05.05.2024	виконано
7	Аналіз та впорядкування результатів	12.05.2024	виконано
8	Оформлення пояснювальної записки та презентації для демонстрації	19.05.2024	виконано

Студент Максим МАКІТРУК

Керівник Юрій СЕЛІН

РЕФЕРАТ

Дипломна робота: 60 с.,27 рис.,8 табл., 3дод., 15 джерел.

ПРОГНОЗУВАННЯ, ФІНАНСОВІ ЧАСОВІ РЯДИ, ARIMA, LSTM, ІНДЕКС S&P 500, КУРС ЗОЛОТА, МАШИННЕ НАВЧАННЯ, СТАТИСТИЧНИЙ АНАЛІЗ

Об’єкт дослідження – Фінансові часові ряди, методи прогнозування фінансових часових рядів.

Предмет дослідження – Методи прогнозування, що базуються на ARIMA та LSTM моделях.

Мета роботи – Провести аналіз методів прогнозування фінансових часових рядів, реалізувати моделі ARIMA та LSTM для прогнозування індексу S&P 500 та курсу золота, а також провести тестування побудованих моделей на історичних даних.

Методи дослідження – методи машинного навчання та статистичного аналізу, авторегресійно-інтегрована модель ковзного середнього (ARIMA) та рекурентні нейронні мережі довгострокової пам'яті (LSTM).

Актуальність – у сучасному світі економічних нестабільностей та фінансових криз, прогнозування фінансових часових рядів є важливою задачею для інвесторів та аналітиків. Використання сучасних методів машинного навчання дозволяє отримувати більш точні прогнози, що сприяє прийняттю обґрунтованих рішень та зменшенню ризиків.

Результати роботи – результати показують, що модель LSTM забезпечує високу точність прогнозування фінансових часових рядів з низькими показниками похибок. Модель ARIMA також демонструє хорошу ефективність, але поступається LSTM за точністю прогнозів. Використання цих моделей дозволяє

підвищити рентабельність інвестицій та приймати більш обґрунтовані рішення на фінансових ринках.

Шляхи подальшого розвитку предмету дослідження – збільшити кількість параметрів, що враховуються в моделях, дослідити можливості застосування методів прогнозування для інших фінансових інструментів, а також адаптувати моделі для прогнозування в умовах високої волатильності ринків.

ABSTRACT

Diploma work: 60 p., 27 fig., 8 tabl., 3 appendixes, 15 references.

FINANCIAL TIME SERIES FORECASTING, ARIMA, LSTM, MACHINE LEARNING, STOCK MARKET ANALYSIS, GOLD PRICES, INVESTMENT STRATEGIES

Object of research – Financial time series forecasting.

Subject of study – Forecasting methods based on ARIMA and LSTM models.

The purpose of the work – To analyze the subject of research, implement ARIMA and LSTM models for forecasting the S&P 500 index and gold prices, and test the built models on historical data.

Methods of research – Machine learning and statistical analysis methods: AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) neural networks.

Actuality – In the modern world of economic instability and financial crises, forecasting financial time series is a crucial task for investors and analysts. Using modern machine learning methods allows for more accurate forecasts, facilitating well-informed decisions and risk reduction.

The results of the work – The results show that the LSTM model provides high accuracy in forecasting financial time series with low error rates. The ARIMA model also demonstrates good efficiency but is less accurate than LSTM. Using these models enhances investment profitability and supports more informed decision-making in financial markets.

Ways of further development of the subject of research – To increase the number of parameters considered in the models, explore the applicability of forecasting methods for other financial instruments, and adapt the models for forecasting under high market volatility conditions.

ЗМІСТ

СКОРОЧЕННЯ

ARIMA – autoregressive integrated moving average

RNN – Recurrent neural network (рекурентні нейронні мережі)

MA – ковске середнє

AR – авторегресія

I – інтегрування

ML – machine learning (машинне навчання)

RF – Random Forests (випадкові ліси)

GBM – Gradient Boosting Machines

LSTM – Long short-term memory

ACF – Auto-correlation function (Автокореляційна функція)

PACF – partial autocorrelation function (часткова автокореляційна функція)

AIC – Akaike information criterion

BIC – Bayesian information criterion

SIC – Schwarz criterion

MAE – Mean Absolute Error (середня абсолютна похибка)

MAPE – Mean Absolute Percentage Error (сер. абсолютна відносна похибка)

RMSE – Root Mean Squared Error (корінь сер. квадратичного відхилення)

ВСТУП

Ми живемо у світі економічних війн та маніпуляцій, де багаті можуть маніпулювати курсом валют та облігацій задля свого збагачення. У сучасному світі, де одне висловлювання впливового політика або ухвалення нового закону може значно вплинути на курси валют протягом багатьох років. Саме тому питання фінансової грамотності та розуміння базових принципів статистики є необхідністю для будь якої сучасної людини, що хоче розвиватись, зберігати чи примножувати свої статки.

Моя робота виходить за рамки простого опису алгоритмів, адже вона зосереджується на прогнозуванні фінансових даних, які представлені у вигляді часових рядів. Фінансові часові ряди – це послідовності числових даних, зібраних в регулярні інтервали часу, таких як ціни акцій, курси валют, відсоткові ставки та інші економічні індикатори. Прогнозування таких рядів має вирішальне значення для інвесторів, аналітиків та політиків, оскільки дозволяє приймати обґрунтовані рішення щодо майбутніх тенденцій.

Різні учасники ринку знаходять значну цінність у прогнозуванні фінансових часових рядів. Інвестори можуть використовувати прогнози для прийняття рішень щодо купівлі або продажу активів, зменшення ризиків та максимізації прибутків. Аналітики використовують ці прогнози для розробки стратегій управління портфелями та оцінки ризиків. Для політиків та центральних банків ці прогнози можуть бути корисними для прийняття рішень щодо монетарної політики, управління інфляцією та підтримки стабільності фінансових ринків.

Таким чином, використання методів штучного інтелекту для прогнозування фінансових часових рядів відкриває нові можливості для точнішого та надійнішого прогнозування фінансових показників. Це дозволяє учасникам ринку краще орієнтуватися в економічних умовах, приймати обґрунтовані рішення та ефективно управляти ризиками.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальні методи прогнозування

Якщо маємо справу з фінансовими часовими рядами, особливо якщо вони відображають динаміку цін акцій, курс валют відносне одне одного, цін на товари або індекси ринку, існують кілька підходів до прогнозування, які є особливо ефективними для таких типів даних.

1. ARIMA моделі часто використовуються для прогнозування фінансових часових рядів. Вони дозволяють моделювати складні залежності між значеннями ряду та прогнозувати його майбутні значення.

2. Методи машинного навчання, такі як Random Forests, Gradient Boosting Machines, нейронні мережі (зокрема LSTM), можуть бути ефективними для прогнозування фінансових часових рядів, особливо коли вони мають складну структуру та велику кількість факторів.

3. У ході моїх досліджень з системного аналізу було виявлено, що різні методи з різною точністю прогнозують майбутні значення, тому варто розглядати прогнозування як комбінацію методів через їх точність та час прорахунку, тому часто ефективним підходом є комбінація різних моделей, таких як ARIMA з GARCH або ARIMA з моделями машинного навчання. Це дозволяє краще врахувати складність та особливості фінансових часових рядів.

4. Окрім модельних підходів, також можна використовувати прогнозування на основі індикаторів та фундаментальних факторів, таких як показники технічного аналізу, макроекономічні показники, новини та інші. Цей метод хоч і є дуже важливим але у концепції мого дослідження не є ключовим через те що оцінити важливість тих чи інших подій та показників можуть експерти, а я не маю такої великої кількості досвіду.

Важливо також враховувати, що фінансові часові ряди часто мають велику кількість шуму та неочікувані зміни, тому для їх прогнозування важливо ретельно виконувати аналіз даних, використовувати методи перевірки та валідації прогнозів, а також бути готовими до корекції прогнозів відповідно до нових інформаційних потоків.

1.2 Пошук математичних схем

Основною перевагою використання методів прогнозування фінансових часових рядів є можливість обробляти та аналізувати великі обсяги даних, це дуже важливо у фінансовій сфері, адже тут швидкість та точність отримання інформації мають вирішальне значення, таким чином, математики, працюючи на знаходження такої інформації розраховують на винагороду, що тільки більше спонукає досліджувати цю тематику.

Можна з упевненістю сказати що формули та схеми обміну акцій, валют та облігацій для заробітку уже існують, але цінні вони саме тому що є секретними, адже при втраті конфіденційності та наборі популярності подібні схеми стають не робочими через ажіотаж. Про такі історії знімають фільми та пишуть художні твори, але це тільки підтверджує правдивість і реальність існування схем.

Таким чином ще з кінця 20го століття математики з усього світу намагаються знайти шляхи передбачення ринку й продати свої ідеї інвесторам за неймовірні гроші, але виходить це лише у тих, хто знайдуть шляхи швидше інших та такі, які будуть найнадійнішими з пропозицій.

1.3 Вибір точності та характеристик моделі

Застосування прогнозування фінансових часових рядів до фондового ринку має свої виклики. Точність прогнозів зазвичай знижується з довшими горизонтами прогнозування, а ефективність моделей може варіюватися для різних ринків і секторів. Крім того, успіх цих моделей залежить від якості даних і правильного вибору характеристик. Вибір функцій і попередня обробка даних є критично важливими етапами в процесі моделювання, оскільки вони допомагають зменшити розмірність даних і видалити нерелевантні характеристики, підвищуючи тим самим точність прогнозування.

Простими словами, якщо вводити багато змінних до розгляду, то пошук тенденцій та залежностей разом із побудовою графіків та відновленням функціональних залежностей будуть забирати кратно більше часу та обчислювальних потужностей, а також кількість параметрів часто збільшують множину та величину погрішностей прогнозування через не правильно розраховані коефіцієнти чи недооцінену важливість факторів, зазвичай для оцінки факторів залучають команди експертів, але навість так більшість параметрів прибирається і до основних підрахунків вираховується погрішність, яка складається з можливого впливу факторів що не були включені до системи.

Незважаючи на ці виклики, методи прогнозування фінансових часових рядів показали значний потенціал у прогнозуванні тенденцій на фондовому ринку. Такі методи, як рекурентні нейронні мережі (RNN) і моделі випадкових лісів, особливо відзначилися своєю ефективністю у методах штучного інтелекту. RNN, здатні враховувати часові залежності даних, добре підходять для моделювання даних часових рядів, таких як курси акцій. Моделі випадкових лісів, що використовують алгоритми на основі дерева рішень, продемонстрували точність у прогнозуванні фондового ринку шляхом об'єднання результатів кількох моделей [1].

Такі програми штучного інтелекту часто базуються саме на моделі ARIMA, що добре себе зарекомендувала своєю гнучкістю та практичністю у застосуванні з використанням сучасних комп'ютерів. Таким чином, можна зробити висновок, що так само, як побудова моделей у другій половині 20го сторіччя зробили великий крок у прогнозуванні даних, так і новітні методи штучного інтелекту дають величезний поштовх у можливостях та ефективності прогнозуванні фінансових даних у 21му сторіччі. Так само і технологія квантових комп'ютерів стане вирішальним кроком у прогнозуванні і потенційно може зробити передбачення будь яких процесів загально доступним і простим для користування

Сьогодні є тенденція щодо залучення методів штучного інтелекту до математики та статистики, так інтеграція методів штучного інтелекту у фінансове прогнозування є значним кроком вперед в аналізі фондового ринку. Ці методи пропонують більш складний, керований даними підхід до розуміння та прогнозування ринкової поведінки. Оскільки технологія штучного інтелекту продовжує розвиватися, її застосування у фінансовому прогнозуванні, ймовірно, стане більш досконалим, відкриваючи нові можливості для досліджень та практики аналізу фондового ринку [1].

1.4 Квантові комп'ютери

Квантові обчислення стали новаторською технологією, яка змінює ландшафт фінансового прогнозування. Завдяки принципам квантової механіки, квантові комп'ютери можуть обробляти величезні обсяги складних даних з безпрецедентною швидкістю, що підвищує точність та ефективність прогнозування у фінансовому секторі. Традиційні методи прогнозування часто не можуть впоратися з великим обсягом і складністю ринкових даних, тоді як

квантові обчислення вирішують ці проблеми за допомогою кубітів, які виконують обчислення, неможливі для класичних комп’ютерів.

Ця зміна парадигми в обчислювальній потужності створить революцію у фінансовому прогнозуванні, дозволяючи аналітикам і дослідникам відкривати нові горизонти в аналізі даних і прогнозуванні. Квантові комп’ютери здатні одночасно виконувати численні обчислення, що дозволяє створювати більш точні та складні моделі, які виявляють приховані закономірності та тенденції у фінансових даних. З розвитком і підвищенням доступності квантових обчислень фінансова індустрія зможе значно виграти від розширених можливостей і нових ідей, які приносить ця передова технологія.

Але не зважаючи на всі новітні перспективи, на сьогоднішній день квантове обладнання ще недостатньо потужне для забезпечення реальних покращень або проведення переконливих масштабних тестів. Однак підвищення продуктивності можна досягти, перетворюючи ці квантові ідеї на класичні рішення машинного навчання (ML). Із вдосконаленням квантового апаратного забезпечення очікується, що ці методи працюватимуть швидше та будуть ще ефективнішими на квантових комп’ютерах. Загальний характер методів робить їх придатними для різноманітних випадків використання у фінансовій сфері та за її межами, хоча вони повинні бути адаптовані до конкретних наборів даних і завдань. Тому перспективи величезні, але сам розвиток людства та комп’ютерів ще не дозволить нам у повному обсязі осягнути та спробувати ці можливості.

1.5 Висновок до розділу 1

Аналіз часових рядів є вельми корисним інструментом для прогнозування, що відкриває підхід до розробки різноманітних методів, таких як авторегресійно-інтегрована модель з ковзним середнім (ARIMA). У світі фінансів, де кожен рух ринку може мати величезний вплив на прибуток та втрати, такі моделі стали

неоціненним інструментом для аналізу цінових динамік та прогнозування майбутніх трендів.

Фінансові ринки завжди були динамічними та непередбачуваними, але завдяки ARIMA, LSTM та подібним моделям, аналітики та трейдери здобули засоби для аналізу та передбачення цінових рухів. Проте, з появою методів машинного навчання, таких як рекурентні нейронні мережі (RNN) та випадкові ліси, підходи до прогнозування стали більш різноманітними та ефективними.

Сучасні математичні методи не тільки допомагають передбачити ринкові тенденції, але й дозволяють розробляти стратегії для здійснення успішних інвестиційних рішень. Аналіз фінансових часових рядів стає все більш складним завдяки зростанню обсягу даних та розвитку технологій.

Одним із ключових напрямків у розвитку фінансового прогнозування є квантові обчислення. Вони відкривають нові горизонти, дозволяючи аналізувати величезні обсяги даних з небаченою швидкістю та точністю. Незважаючи на обмеження апаратного забезпечення, використання квантових принципів у класичних методах машинного навчання вже показало потенціал для значного покращення продуктивності та точності прогнозів.

З кожним днем фінансова індустрія стає все більш залежною від аналізу даних та передбачення майбутніх ринкових рухів. Розвиток методів аналізу, використання новітніх технологій та постійний пошук нових підходів дозволяють тримати руку на пульсі ринку та забезпечують конкурентну перевагу тим, хто може швидко та ефективно аналізувати та передбачати його рухи.

РОЗДІЛ 2. ТЕОРЕТИЧНИЙ ОПИС МЕТОДІВ ПРОГНОЗУВАННЯ ТА ГОЛОВНИХ ПАРАМЕТРІВ І МЕТРИК ДЛЯ ПРОГНОЗУВАННЯ ФІНАНСОВИХ ЧАСОВИХ РЯДІВ

2.1 Опис моделі прогнозування ARIMA

AutoRegressive Integrated Moving Average (ARIMA) є популярним інструментом для аналізу та прогнозування часових рядів. Її використання охоплює різні сфери, у цій роботі нас цікавить фінансове прогнозування, де вона застосовується для передбачення цін на акції, обсягу продажів і економічних показників.

ARIMA поєднує три основні компоненти: авторегресію (AR), інтеграцію (I) та ковзне середнє (MA), розпишимо компоненти детально.

1. Авторегресія (AR) – використовує залежність між поточним значенням часового ряду і його попередніми значеннями (2.1). Параметр p визначає порядок AR, тобто кількість попередніх значень, які використовуються для прогнозування поточного значення. Формула AR процесу:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t, \qquad (2.1)$$

де $y_t -$ поточне значення часового ряду;

$c -$ константа;

ϕ_i – коефіцієнти моделі;

ϵ_t – білий шум.

2. Інтегрування (I) – використовується для перетворення часового нестаціонарного ряду в стаціонарний шляхом різницювання. Параметр **d** визначає порядок інтеграції, тобто кількість разів, які потрібно провести різниціювання, щоб досягти стаціонарності. Різницювання першого порядку:

$$y'_t = y_t - y_{t-1} \tag{2.2}$$

3. Ковзне середнє (MA) – використовує залежність між поточним значенням часового ряду і попередніми випадковими шумами (помилками прогнозування). Параметр q визначає порядок MA, тобто кількість попередніх шумів, які використовуються для прогнозування поточного значення. Формула MA-процесу:

$$y_t = c + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{2.3}$$

де $\theta_i -$ коефіцієнти моделі.

Зібравши все отримуємо :

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{2.4}$$

де $y_t -$ стаціонарний ряд.

Порядок моделі ARIMA визначається як ARIMA(p, d, q):

1) p – порядок авторегресії (кількість попередніх значень);

2) d – порядок інтеграції (кількість необхідних різниць);

3) q – порядок ковзного середнього (кількість попередніх помилок).

Вибір параметрів порядку моделі ARIMA(p,d,q) є найважливішим кроком у побудові моделі ARIMA. Це зазвичай робиться за допомогою аналізу автокореляційної функції (ACF) та часткової автокореляційної функції (PACF), а також критерію інформації Акаіке (AIC) або Байєсового інформаційного критерію (BIC) для оцінки моделей.

2.2 Методи визначення параметрів для моделі ARIMA

У цій роботі задля встановлення параметрів для моделі ARIMA я користуюсь двома методами, а саме методом автоматичного підбору параметрів

та методом аналізу графіків ACF та PACF (повний та частковий графіки автокореляції). Також для встановлення чи є дані стаціонарними варто використовувати тест Дікі-Фуллера, він використовується для перевірки гіпотези про наявність одиничного кореня в часових рядах, тобто для визначення, чи є часовий ряд стаціонарним. Тест ґрунтується на моделі авторегресії.

Тобто основною метою тесту Дікі-Фуллера є перевірка гіпотези 0 проти гіпотези 1:

1) H_0 – часовий ряд має одиничний корінь (нестабільний);

2) H_1 – часовий ряд стаціонарний.

Математична модель

Розглядається авторегресійна модель першого порядку (AR(1)):

$$y_t = \rho y_{t-1} + \epsilon_t \tag{2.5}$$

де $y_t -$ значення часового ряду в момент t;

$\rho -$ коефіцієнт авторегресії;

$\epsilon_t -$ випадкива похибка (білий шум).

Для перевірки одиничного кореня ми перетворимо модель в альтернативну форму:

$$y_t - y_{t-1} = (\rho - 1)y_{t-1} + \epsilon_t \tag{2.6}$$

Позначивши $\delta = \rho - 1$, отримаємо:

$$\Delta y_t = \delta y_{t-1} + \epsilon_t \tag{2.7}$$

де $\Delta y_t = y_t - y_{t-1}$.

Для тестування гіпотези $H_0: \delta = 0$ проти $H_1: \delta < 0$ використовують статистику t-статистики:

$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})} \tag{2.8}$$

де $\hat{\delta} -$ оцінка параметра $\delta, SE(\hat{\delta}) -$ стандартна похибка $\hat{\delta}$.

У роботі користуюсь таким правилом – якщо тест показує значення більше за 0,05 , то ряд не можна вважати стаціонарим і варто проводити різниціювання даних. Якщо після виконання різниціювання значення все ще перевищує 0,05 , то варто провести його повторно, але цей процес враховує різниціювання до другого

порядку, адже дослідження показало що при різниціюванні вище другого порядку для фінансових даних не дає інформативних прогнозувань.

Метод перебору має на меті мінімізувати параметри які відповідають за якість моделі, для роботи були обрані критерій AIC (критерій Акайке) та критерій BIC (критерій Баєсівської інформації).

Критерій Акаіке (Akaike Information Criterion, AIC) використовується для порівняння якості статистичних моделей. Він враховує як точність моделі, так і її складність, щоб уникнути перенавчання. У вигляді формули AIC визначається:

$$AIC = 2k - 2ln(L) \quad (2.9)$$

де $k -$ кількість параметрів моделі;

$L -$ максимальне значення функції правдоподібності моделі.

Критерій Байєсівської інформації (Bayesian Information Criterion, BIC) також використовується для порівняння моделей, але він сильніше штрафує за складність моделі порівняно з AIC. У вигляді формули BIC визначається:

$$BIC = kln(n) - 2ln(L) \quad (2.10)$$

де $k -$ кількість параметрів моделі;

$n -$ розмір вибірки(кількість спостережень);

$L -$ максимальне значення функції правдоподібності моделі.

На практиці це виглядає як перебір параметрів порядку моделі ARIMA(p,d,q) з значень: {0,1,2,3,4,5} для p та q, та {0,1,2} для d, той набір що виявляється найкращим за обома критеріями і приймається основним.

Метод підбору вручну залучає використання автокореляційної функції (ACF) та часткової автокореляційної функції (PACF), що допомагає виявити структуру часових рядів і вибрати оптимальні параметри моделі. Цей підхід дозволяє створювати моделі, що точно відображають внутрішню динаміку даних.

На основі даних з літератури та дослідження склав алгоритм вибору параметрів порядку.

1. З огляду на графік даних потрібно зрозуміти чи є дані стаціонарними, аби перевірити свої здогатки потрібно скористатись тестом Дікі-Фуллера, якщо тест видає значення що перевищує 0,05 , то варто збільшити

параметр d можелі ARIMA на одиницю і повторити тест, при результаті меншого за 0,05, можна відкинути нульову гіпотезу та вважати часовий ряд стаціонарним.

2. Після визначення параметру d потрібно побудувати графіки ACF та PACF та з їх результату визначити параметри p та q за таким принципом – якщо на графіку PACF є велике значення на лагу p,а далі більше не має, а ACF графік спадає поступово, то варто розглянути модель з параметрами (p,d,q)=(p,d,0). Якщо графік ACF має велке значення на лагу q, але не далі, а PACF спадає поступово, то варто обрати модель з параметрами (p,d,q)=(0,d,q).

3. Якщо графіки не дають такого чіткого визначення структури, то варто застосувати обидва параметри та задавати значення відповідно структурі даних.

2.3 Економічні параметри та метрики для прогнозування

При прогнозуванні фінансових часових рядів можна використовувати безліч параметрів, так наприклад рівень інфляції та відсоткові тавки центрального банку можуть безпосередньо впливати та фондовий ринок та корегувати ціни на акції. Задаля виконання точного та більш комплексного прогнозування можна завжди збільшити кількість змінних та шукати між ними залежності, аби краще аналізувати тенденції та тренди, аби прогнози були точнішими. У цій роботі я фокусуватимусь лише на безпосередніх параметрах що відображають вартісь активів, аби не перенавантажувати обчислювання.

При вдосконаленні моделей можна використати наступні економічні параметри.

1. ВВП є основним показником економічної активності та здоров’я економіки. Його зростання або спад може впливати на курси валют, ціни акцій та інші фінансові ринки. Для аналізу графіків цін на золото та індекс S&P 500, зміни у ВВП можуть бути використані для оцінки загального економічного стану.

2. Високий рівень безробіття може свідчити про слабку економіку, що може негативно вплинути на фінансові ринки. Зміни у рівні безробіття можуть бути корисними для прогнозування ринкових тенденцій.

3. Рівень інфляції відображає зростання цін на товари та послуги. Висока інфляція може призвести до підвищення відсоткових ставок, що впливає на курси валют та ринки облігацій. Інфляційні показники можуть бути використані для коригування прогнозів цін на золото та індекс S&P 500.

4. Політика центральних банків щодо відсоткових ставок безпосередньо впливає на фінансові ринки. Зниження ставок може стимулювати економічне зростання, тоді як їх підвищення може охолодити економіку. Відсоткові ставки є важливими для прогнозування цін на активи.

5. Фінансовий стан уряду, включаючи дефіцит чи профіцит бюджету, впливає на економічну стабільність та довіру інвесторів. Цей параметр може бути використаний для оцінки загальної економічної ситуації.

6. Торговий баланс відображає різницю між експортом та імпортом країни. Профіцит торгового балансу зазвичай сприяє економічному зростанню, тоді як дефіцит може призвести до зниження ВВП. Торговий баланс може впливати на курси валют та ринки активів.

Для оцінки ефективності прогнозування у цій роботі я користуюсь трьома метриками оцінки моделей.

1. MAE (Mean Absolute Error) – вимірює середню абсолютну похибку між прогнозованими та фактичними значеннями. Це дає уявлення про середню похибку прогнозів.

2. MAPE (Mean Absolute Percentage Error) – вимірює середню абсолютну відносну похибку між прогнозованими та фактичними значеннями. Це корисно для оцінки точності моделей у відсотковому вираженні.

3. RMSE (Root Mean Squared Error) – вимірює корінь середнього квадратичного відхилення між прогнозованими та фактичними значеннями. Це показник, що акцентує увагу на великих похибках, роблячи його корисним для виявлення великих відхилень у прогнозах.

2.4 Опис принципів методу передбачення моделі LSTM

LSTM – це особливий вид рекурентної нейронної мережі (RNN), який був створений для подолання обмежень традиційних RNN. Основна перевага LSTM полягає в її здатності зберігати інформацію протягом тривалих проміжків часу завдяки механізму гейтів.

Опишимо модель LSTM у математичному та прикладному контексті, модель складається з таких компонентів.

1. Комірки пам'яті (Memory Cells) – основний елемент, який зберігає інформацію.

2. Вхідний гейт (Input Gate) – контролює, яка нова інформація буде додана до комірки пам'яті. Представлений у вигляді формули:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (2.11)$$

$$\widetilde{C_t} = tanh(W_C * [h_{t-1}, x_t] + b_C) \quad (2.12)$$

де $\widetilde{C_t} -$ кандидат на новий стан клітини;

i_t – вектор вхідних ворт.

3. Забуваючий гейт (Forget Gate) – вирішує, яку інформацію з комірки пам'яті потрібно забути. Представлений у вигляді формули:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (2.13)$$

де $f_t -$ Забувальний вектор;

$W_f -$ ваги;

$h_{t-1} -$ попередній прихований стан;

$x_t -$ поточний вхід;

$b_f -$ зсув.

4. Вихідний гейт (Output Gate) – визначає, яку частину інформації з комірки пам'яті буде використано як вихід. Представлений у вигляді формули:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (2.14)$$

$$h_t = o_t * tanh(C_t) \quad (2.15)$$

де $o_t -$ вхідний вектор;

$h_t -$ поточний прихований стан.

5. Оновлення стану клітини, представлена у вигляді формули:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (2.16)$$

де $C_t -$ поточний стан клітини.

LSTM використовує ці рівняння для регулювання інформаційного потоку через клітину, зберігаючи важливу інформацію та забуваючи нерелевантну, що допомагає запобігти проблемі згасання градієнта та зберігати довгострокові залежності.

Ця архітектура дозволяє LSTM ефективно обробляти часові ряди та передбачати майбутні значення, враховуючи довготривалі залежності в даних.

2.5 Прогнозовані величини та їх опис

Для проведення дослідження та порівняння методів прогнозування часових рядів я обрав два ключові фінансові показники – індекс S&P 500 та курс золота. Ці два показники є важливими маркерами економічного стану та ринкових настроїв.

S&P 500 є одним із найвідоміших і найбільш широко використовуваних фондових індексів у світі. Він відстежує ринкову капіталізацію 500 найбільших публічних компаній Сполучених Штатів. Індекс використовується як індикатор фінансового стану економіки та є одним із головних орієнтирів для інвесторів та аналітиків. Його вартість коливається залежно від різноманітних факторів, включаючи економічні показники, політичні події та настрої інвесторів.

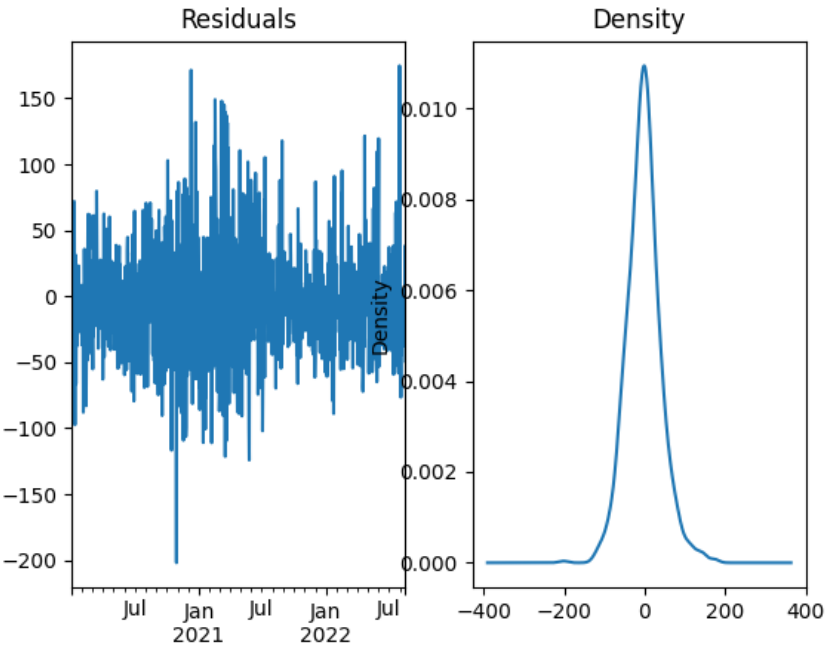
З іншого боку, ціна золота є показником вартості одного з найбільш стабільних і надійних активів у світі. Золото традиційно вважалося надійним

притулком для інвесторів у часи економічної невизначеності. Його вартість залежить від багатьох факторів, включаючи економічні, політичні та соціальні обставини. Відстеження цін на золото допомагає зрозуміти глобальні економічні тенденції та настрої ринку.

2.6 Перевірка моделей на адекватність

Після побудови моделі важливо переконатися в її адекватності. Для цього я використовую декілька підходів, зокрема графічний аналіз залишків та тест Льюнга-Бокса. Графічний аналіз залишків дозволяє мені переконатися, що залишки моделі не мають видимих структур, таких як тренди або автокореляції. Тест Льюнга-Бокса підтверджує відсутність автокореляції в залишках, що є додатковим підтвердженням адекватності моделі.

На прикладі прогнозування індексу S&P 500 проводжу демонстрацію цих дій, накладаю модель та дані і будую графік викидів та відхилень аби зрозуміти чи можна вважати їх білим шумом і випадковими збуреннями. Будую графік залишків та будую графік його розподілення:



Рисинок 2.1 – Графік викидів та відхилень для значень S&P 500

З рисунку видно, що відхилення зосереджені навколо нуля, що є хорошим знаком і приводом вважати модель адекватною і якісною.

2.7 Висновок до розділу 2

У цьому розділі було детально розглянуто теоретичні основи методів прогнозування фінансових часових рядів, зокрема моделі ARIMA та LSTM, а також головні змінні, що впливають на фінансові ринки.

Було показано, що модель ARIMA (AutoRegressive Integrated Moving Average) є ефективним інструментом для прогнозування часових рядів завдяки своїй здатності враховувати як авторегресійні залежності, так і ковзні середні, а також використовувати різницювання для досягнення стаціонарності. Викладено математичні основи кожного компонента моделі та методи вибору параметрів за допомогою ACF, PACF, AIC та BIC.

Для перевірки стаціонарності часових рядів було розглянуто тест Дікі-Фуллера. Цей тест дозволяє визначити наявність одиничного кореня в часових рядах, що є критичним для правильного застосування моделі ARIMA.

Розглянуто важливість використання економічних параметрів, таких як ВВП, рівень безробіття, інфляція, відсоткові ставки, дефіцит/профіцит бюджету та торговий баланс, для покращення точності прогнозів. Ці змінні допомагають краще зрозуміти контекст фінансових ринків та зробити більш точні прогнози.

LSTM (Long Short-Term Memory) була описана як потужний інструмент для роботи з часовими рядами, що дозволяє враховувати довготривалі залежності в даних. Було наведено математичну модель LSTM, включаючи вхідні, забувальні та вихідні ворота, а також механізм оновлення стану клітини.

Розглянутий теоретичний матеріал створює міцну основу для практичного застосування методів ARIMA та LSTM у прогнозуванні фінансових часових рядів. У наступних розділах ці моделі будуть застосовані для аналізу реальних фінансових даних з метою порівняння їх ефективності та точності прогнозування.

РОЗДІЛ 3. ПРАКТИЧНЕ ВИКОНАННЯ ПРОГНОЗУВАННЯ ТА ПОРІВНЯННЯ МЕТОДІВ.

Метою цього розділу є дослідження і порівняння двох популярних методів прогнозування часових рядів – ARIMA та LSTM – на прикладі фондового індексу S&P 500 та курсу вартості золота. Розуміння цих методів і їх застосування може значно підвищити точність прогнозів, що має велике значення для аналітиків, інвесторів і дослідників.

Для досягнення цієї мети, необхідно пройти через кілька важливих етапів. Почнемо з обробки даних, що включає збір, очищення та передобробку даних. Це дозволить підготувати якісні вхідні дані для моделей. Очищення даних від аномалій і заповнення пропусків, нормалізація значень та візуалізація дозволяють краще розуміти структуру даних і виявляти приховані закономірності. Сезонна декомпозиція, у свою чергу, допоможе виділити основні компоненти часового ряду, такі як тренд, сезонність і випадкові коливання.

Далі розгляну метод ARIMA, який зарекомендував себе як надійний інструмент для аналізу та прогнозування часових рядів. Вибір правильних параметрів моделі – (p, d, q) – є ключовим для досягнення високої точності прогнозів. Я скористаюсь автокореляційною функцією (ACF) та частковою автокореляційною функцією (PACF), щоб визначити оптимальні значення параметрів, після чого застосую модель ARIMA для прогнозування курсу S&P 500 та золота. Оцінку якості моделей проведу за допомогою метрик точності та графічного аналізу.

Наступним кроком буде розгляд моделі LSTM, яка є однією з передових методів машинного навчання для прогнозування часових рядів. Архітектура LSTM дозволяє ефективно працювати з даними, що мають довгострокові залежності. Я описую налаштування параметрів цієї моделі, процес її тренування на даних про індекс S&P 500 та курс цін на золото, а також проведу оцінку її

якості. Порівняння результатів прогнозування між методами ARIMA та LSTM дозволяє виявити сильні та слабкі сторони кожної з них.

Також приділяю увагу елементам програмного продукту що відображають графіки та метрики потрібні для налаштування моделей та перевірки їх якості прогнозування. Усі рисунки зроблені за допомогою методів програмування мови Python та засновані лише на результатах роботи самої програми.

Завершуючи цей розділ, я провів порівняння методів прогнозування. Оцінка точності моделей буде здійснена на основі відповідних метрик. Порівняльна таблиця та графіки результатів допоможуть продемонструвати відмінності між методами ARIMA та LSTM. Я зважую переваги та недоліки кожного з методів, роблю висновки щодо їх ефективності та супроводжу це рекомендаціями щодо їх використання в залежності від конкретної задачі.

3.1 Збір та обробка даних

Збір даних про індекс S&P 500 і ціни на золото є важливим кроком у нашому дослідженні. Для цього ми використовуємо різноманітні надійні онлайн-ресурси та фінансові платформи для доступу до історичних даних. Серед таких ресурсів виділяються Yahoo Finance, Google Finance та спеціалізовані фондові сайти, такі як Investing.com і Bloomberg. Ці платформи надають точні та актуальні дані, необхідні для аналізу.

Основним методом збору даних є використання API (Application Programming Interface) цих платформ. API дозволяє автоматизувати процес завантаження даних, що забезпечує їх актуальність та точність. Використання API також дозволяє легко інтегрувати дані у наш аналіз та моделі прогнозування.

Далі, за допомогою API або через завантаження CSV файлів, ми отримуємо необхідні дані.

Не зважаючи на це, заради можливості роботи роботи програми у випадках коли оновлення інформації онлайн неможливе, я записую усі потрібні дані до текстових файлів, з яких уже зчитуватиме інформацію програма.

Після завантаження даних ми переходимо до їх обробки. Цей етап включає очищення даних від можливих аномалій, заповнення пропусків та нормалізацію значень. Через актуальність та попит на точну інформацію, дані що я розглядаю є уже обробленими та зручними для користування, адже показують дійсні історичні дані з біржі. Але при роботі з даними що не володіють таким великим попитом, а тому у відкритому доступі не є перевіреними та відкоригованими, процес обробки є критичним для забезпечення точності та надійності прогнозів. Розглянемо ключові методи обробки даних, включаючи очищення, нормалізацію та виявлення викидів.

Очищення даних це один із найважливіших етапів у процесі передобробки. Він включає видалення або коригування неповних, некоректних або невідповідних даних. У фінансових часових рядах можуть бути пропущені значення, шумові дані або дублікати, що можуть суттєво вплинути на результати моделювання. Тому розглянемо найтиповіші проблеми та їх методи вирішення.

Пропущені значення можуть виникати через технічні помилки або недоступність даних у певні періоди. Одним із методів обробки таких даних є видалення рядків з пропусками, але це може призвести до втрати важливої інформації. Альтернативним підходом є заповнення пропущених значень за допомогою методів інтерполяції або середніх значень.

Після того як дані є повними потрібно виявити дублікати та шуми, адже вони у даних можуть виникати через випадкові та систематичні помилки вимірювань. Дублікати записів, які слід видаляти, щоб уникнути викривлення результатів аналізу. Це особливо актуально для великих наборів даних, де дублікати можуть значно вплинути на модель. Для видалення шумів використовують фільтри низьких частот, ковзне середнє або методи згладжування (наприклад, експоненціальне згладжування).

Після того як ми очистили інформацію дуже корисним кроком є нормалізація даних, вона важлива для приведення до одного масштабу економічних показників що ми використовуємо для оцінки якості моделі. У деяких випадках не проведення нормалізації для обчислень може призвести до домінування одних змінних над іншими під час моделювання, у моїй роботі саме цей аспект не є ключовим через оперування лише з одним параметром при прогнозуванні, але при ускладненні моделі або проведенні відновлення функціональних залежностей для знаходження більш точних даних і прогнозів ця проблема є особливо актуальною.

У роботі використовували та тестували різні методи нормалізації.

1. Мінімакс нормалізація – використовується для масштабування значень змінних до діапазону [0, 1] або [-1, 1]. Формула мінімакс нормалізації:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

де $X -$ оригінальні значення;

$X_{min}, X_{max} -$ мінімальне та максимальне значення відповідно.

2. Стандартизація (z-score нормалізація) – використовується для масштабування даних таким чином, щоб вони мали середнє значення 0 та стандартне відхилення 1. Формула стандартизації:

$$X_{std} = \frac{X - \mu}{\sigma} \tag{3.2}$$

де $X -$ оригінальне значення;

$\mu -$ середнє значення змінної;

$\sigma -$ стандартне відхилення.

3. Логарифмічне перетворення – використовується для зменшення асиметрії розподілу даних та згладжування впливу великих значень. Формула логарифмічного перетворення:

$$X_{log} = log(X + 1) \tag{3.3}$$

Це перетворення особливо корисне для фінансових даних, де розподіли часто мають довгий хвіст.

Наступним етапом є виявлення та обробка викидів. Викиди (залишки) – це аномальні значення, які значно відрізняються від інших спостережень у наборі даних. Вони можуть виникати через помилки вимірювань, випадкові події або екстремальні ринкові умови. Для їх виявлення використовую наведені методи.

1. Статистичні методи виявлення викидів – найпоширенішим методом є використання міжквартильного діапазону (IQR) та z-оцінок. Значення, що виходять за межі 1,5 IQR або мають z-оцінки більше 3 (або менше -3), часто вважаються викидами.

2. Візуальні методи – викиди можуть бути виявлені за допомогою графічних методів що є обширно представленими у бібліотеках мови Python, таких як box plot, scatter plot або контрольні графіки (control charts).

3. Методи заміщення викидів – після виявлення викиди можна обробляти різними способами, зокрема видаленням, заміщенням середніми значеннями або медіанами, або використанням більш складних методів, таких як локальне згладжування або інтерполяція.

У результаті можна підвести підсумок, що обробка даних та передобробка є важливими етапами підготовки фінансових часових рядів для моделювання та прогнозування. Очищення даних, нормалізація та виявлення викидів забезпечують точність та надійність побудованих моделей. Правильно підготовлені дані дозволяють максимально ефективно використовувати можливості сучасних методів машинного навчання та методів математичного прогнозування, забезпечуючи високу якість прогнозів.

Варто також зазначити що дані на різних біржах та сайтах подаються у різних форматах, тому варто у програмі забезпечити конвертацію інформації у правильні формати та величини для того щоб зробити можливою роботу з альтернативними ресурсами.

Оброблені дані зберігаються у зручному форматі, наприклад, у своїй роботі я користуюсь Pandas DataFrame у Python, що дозволяє легко маніпулювати ними та проводити подальший аналіз. Легкість маніпулювання є також ключовим

компонентом, адже це зменшує навантаження на обчислювальні потужності та скорочує загальний об’єм програми.

3.2 Візуалізація даних прогнозування

У ході дослідження я буду розглядати 2 набори даних щоб на них візуалізувати результати. Тож представимо графіки поведінки цін на золото за останні 10 років та значення індексу S&P 500.

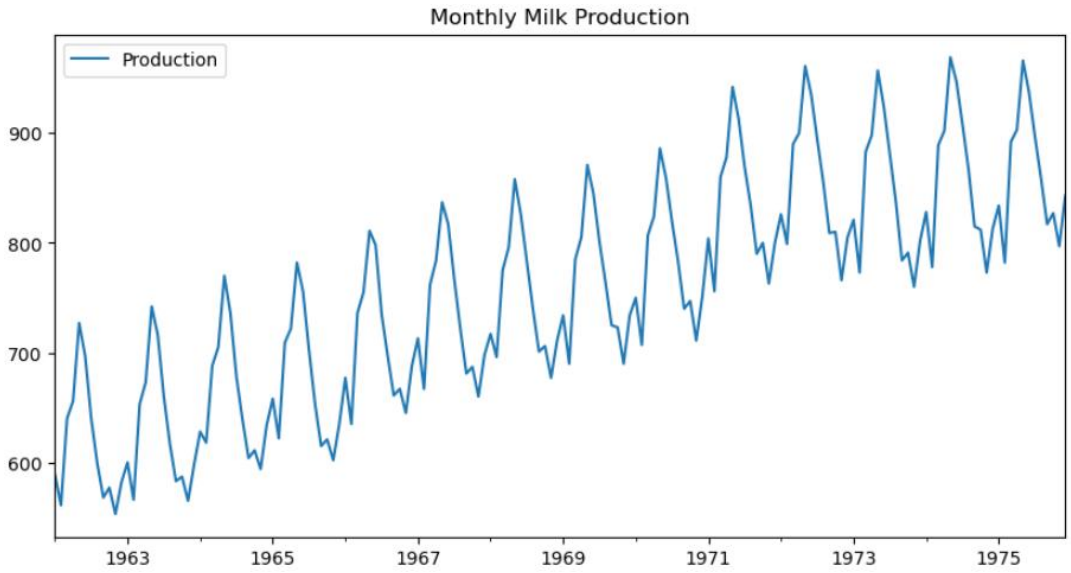
Дані є не стаціонарними та робота з ними може викликати велику кількість похибок, але у методах прогнозування якими я користуюсь передбачені шляхи боротьби з цією проблемою.

Рисинок 3.1 – Графік даних про рух індексу S&P 500

Рисинок 3.2 – Графік даних про ціни золота

Варто зауважити, що найкраще та найточніше прогнозування відбувається саме на сезонних даних, які мають чітко виражений тренд та сезонність. Прикладом таких даних є збір молока з ферми за 10 років чи продаж авіабілетів за 10 років.

Графіки таких даних виглядають так:



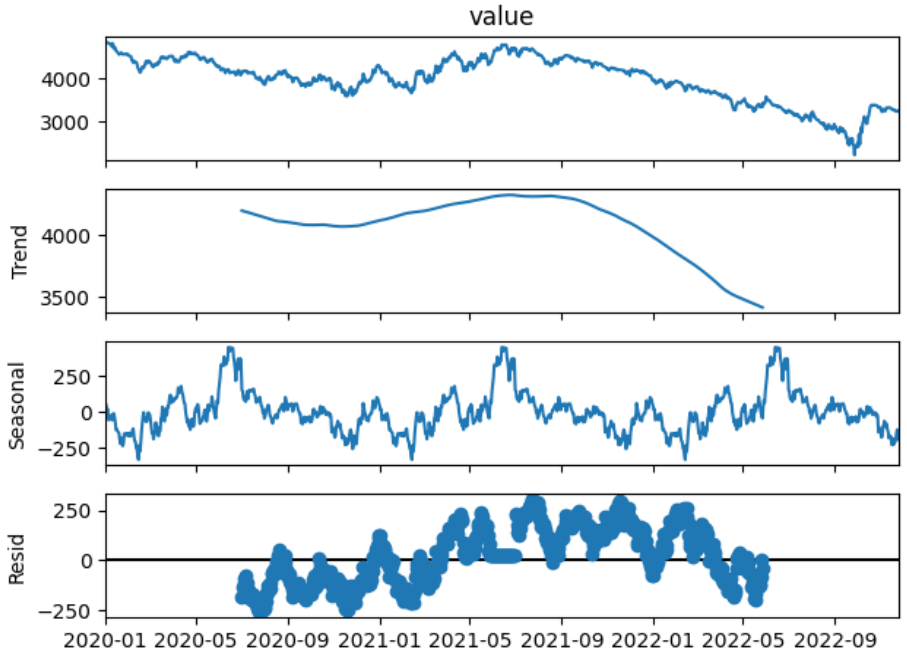
Рисинок 3.3 – Графік даних про збір молока з ферми

Але темою моєї роботи є прогнозування фінансових даних та порівняння результатів, для таких даних існує сезонність та тренд, на які можна спиратись, адже економіки світу мають календарні коливання, економічні цикли та вплив пов’язаний із політичними рішеннями та рішеннями впливових економістів і бізнесменів що часто володіють великими об’ємами фінансів та можуть маніпулювати ними задля досягнення своїх інтересів.

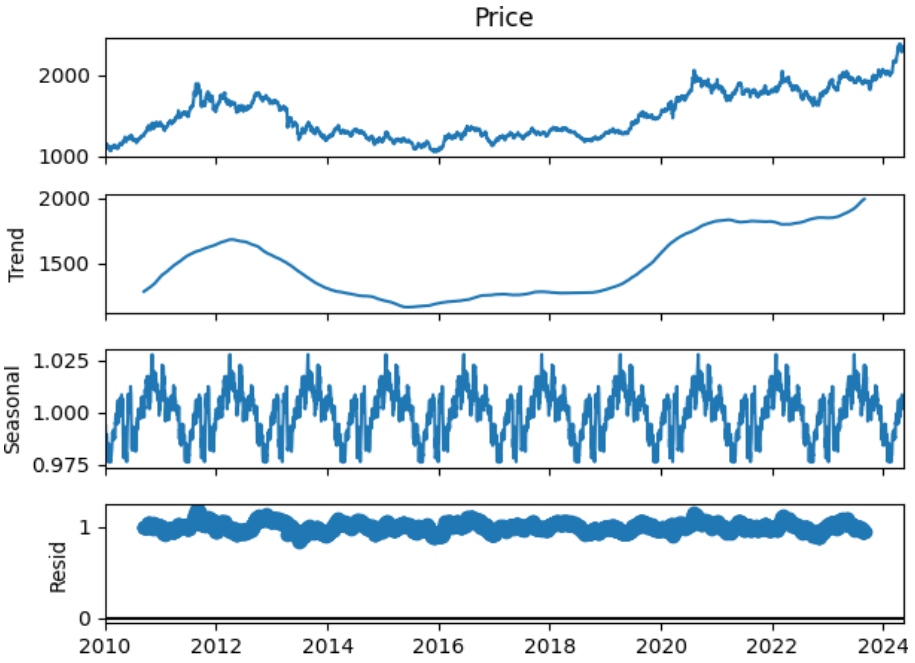
3.3 Сезонна декомпозиція

Для накращого процесу підбору параметрів та визначення довжини прогнозування варто проводити сезонну декомпозицію даних. Цей процес допомагає розділити дані на тренд, сезонність та залишки, що значно спрощує

розуміння структури даних та їх характер. Провівши сезонну декомпозицію отримали такі графіки:



Рисинок 3.4 – Графік декомпозиції даних для індексу S&P 500



Рисинок 3.5 – Графік декомпозиції даних для курсу золота

На графіках представлені:

1) загальний вигляд даних;

2) рух тренду;

3) сезонність;

4) залишки.

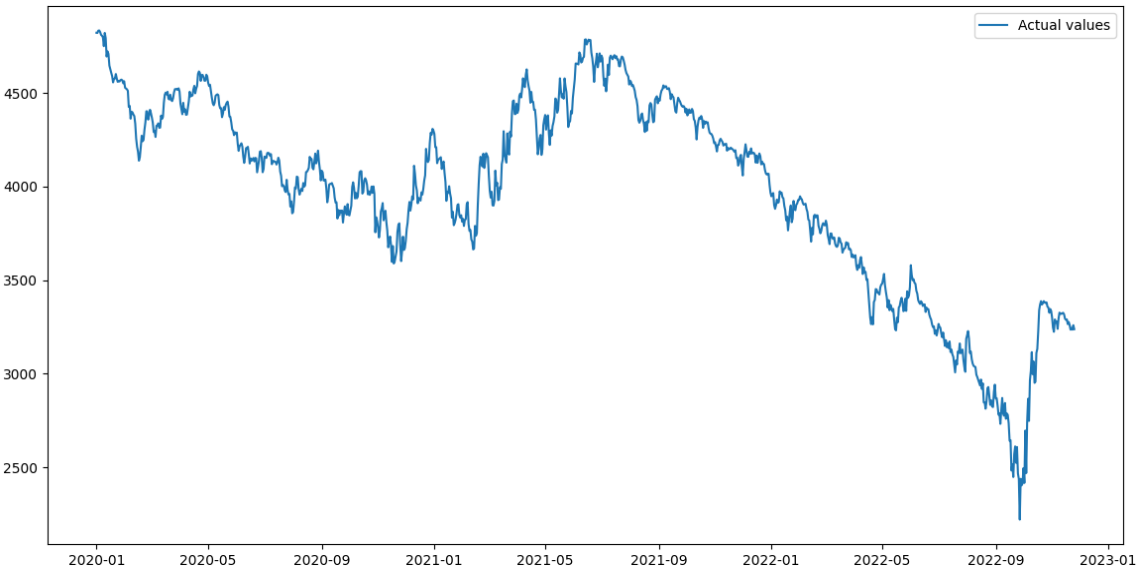
З рисунків видно, що дані мають чіткий тренд, не чітку сезонність та велику кількість викидів, через це можна припустити, що модель ARIMA справиться з задачею прогнозування швидше але гірше за модель LSTM. Також можна зробити спробу покращення сезонної декомпозиції у даних інексу S&P 500, за рахунок скорочення об’єму інформації та видалення завчасно неякісної інформації пов’язаної з початком світових конфліктів. Але моєю метою є спроба прогнозування даних у їх реальному вигляді, тому відповідну ідею можна віднети до можливостей покращення проекту.

3.4 Використання моделі ARIMA

Використання моделі ARIMA для прогнозування цін на золото виявилося цікавим та інформативним процесом, що поєднує в собі як теоретичні знання, так і практичні навички в галузі аналізу часових рядів. Протягом дослідження я стикнувся з декількома важливими аспектами, які варто розглянути детальніше.

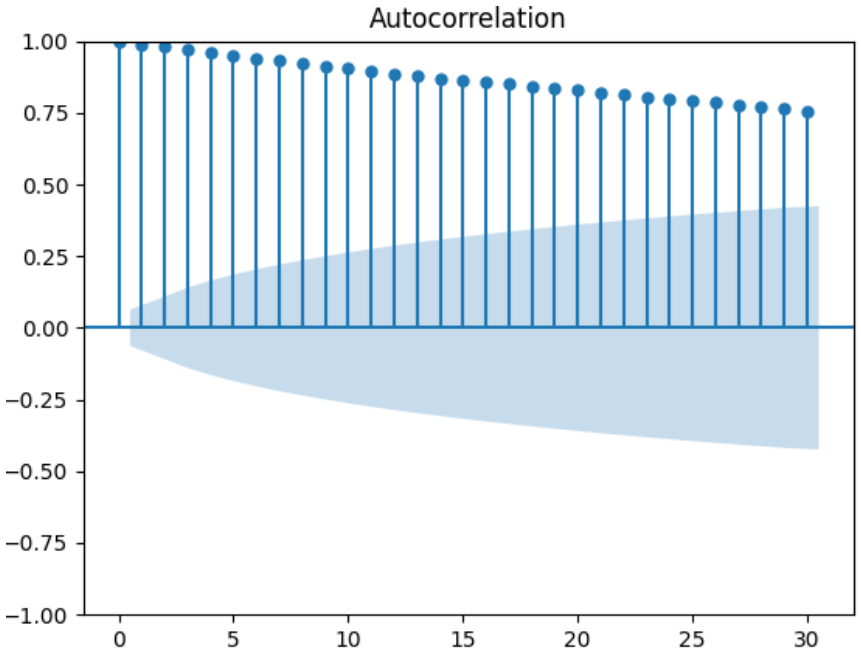
Вибір параметрів ARIMA на прикладі руху індексу S&P 500

Будуємо графік зміни даних:

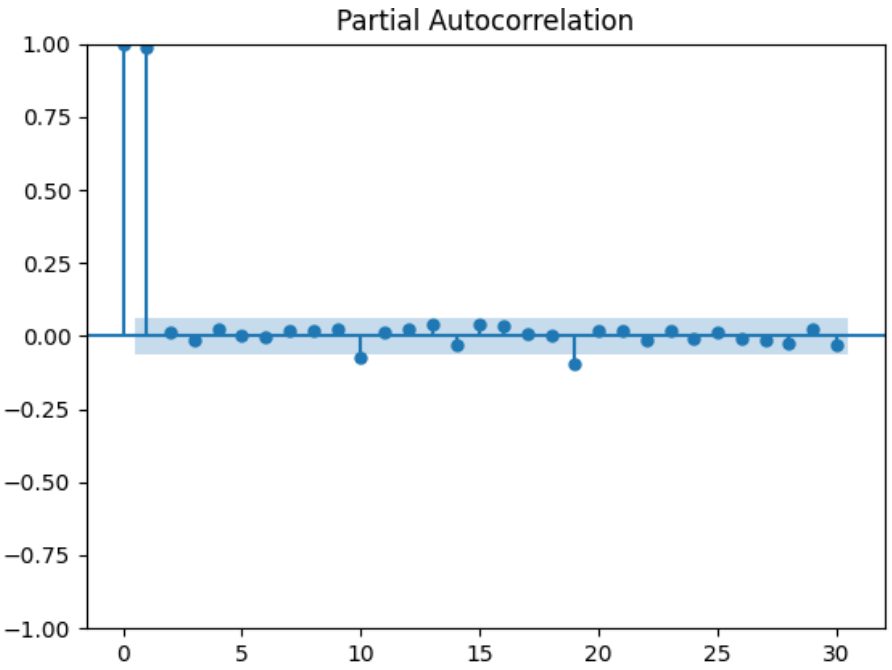


Рисинок 3.6 – Графік руху індексу S&P 500

Будуємо графіки ACF та PACF:



Рисинок 3.7 – Графік ACF для S&P 500 (без різниціювання)



Рисинок 3.8 – Графік PACF для S&P 500 (без різниціювання)

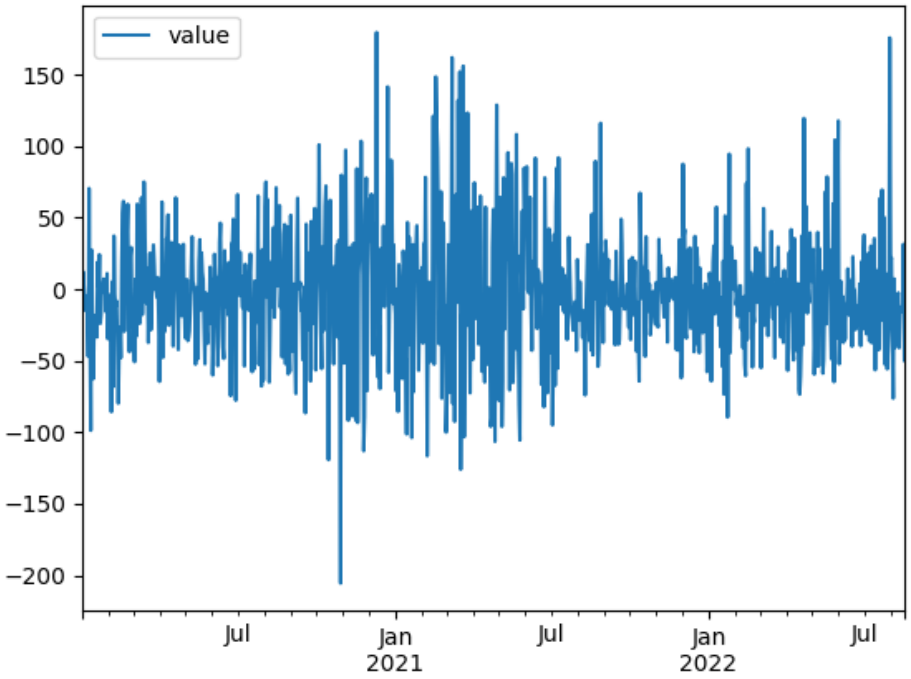
Проводимо тест Дікі-Фуллера

```
1. ADF :  -1.6322599379452831
2. P_Value :  0.46630799407758927
3. Num Of Lags :  9
4. Num Of Observ. used for ADF reges. and Crit. val. calc. :  1050
5. Crit. val. :
     1% :  -3.4365931987759417
     5% :  -2.864296541617536
     10% :  -2.568237690702948
```

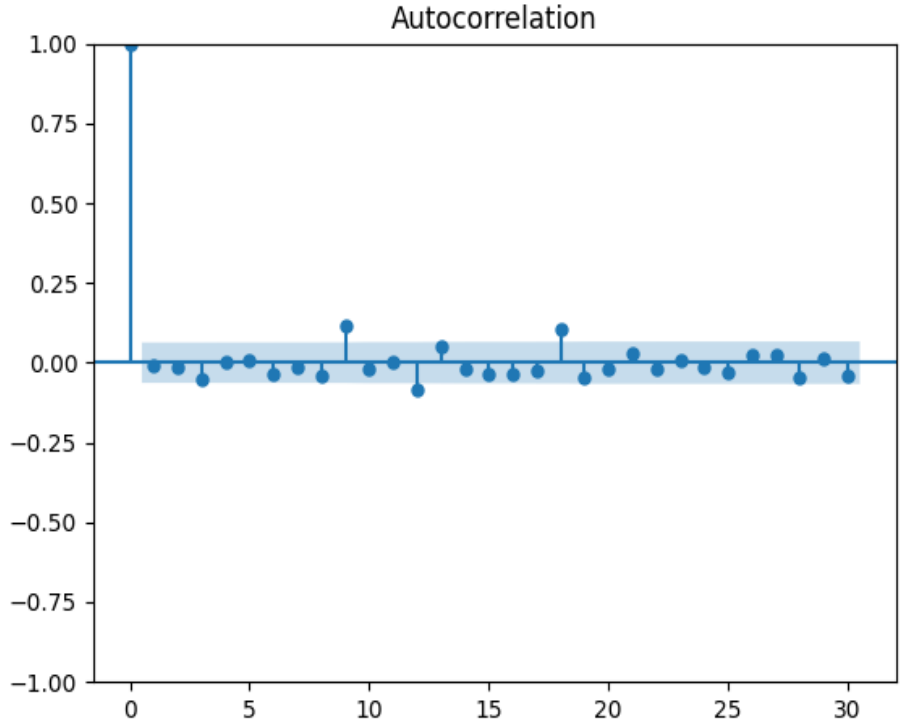
Рисинок 3.9 – Результати тесту для даних S&P500 (без різниціювання)

“P_Value” показує результат тесту, на рисутку 3.10 він рівний 0.466, оскільки це значення перевищує 0,05, то варто провести різниціювання та збільшити параметр d на одиницю.

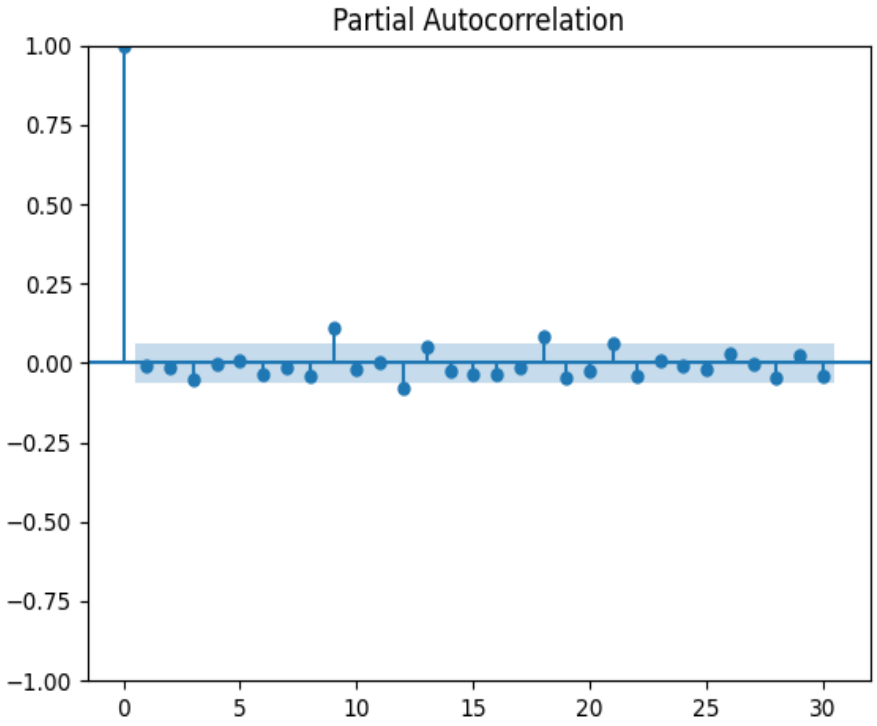
Провівши різниціювання отримали:



Рисинок 3.10 – Графік руху даних S&P 500 (після різниціювання)



Рисинок 3.11 – Графік ACF індексу S&P 500 (після різниціювання)



Рисинок 3.12 – Графік PACF індексу S&P 500 (після різниціювання)

```
1. ADF :  -6.763563342652911
2. P_Value :  2.7531274485120787e-09
3. Num Of Lags :  20
4. Num Of Observ. used for ADF reges. and Crit. val. calc. :  938
5. Crit. val. :
     1% :  -3.4373407098114765
     5% :  -2.8646262040163566
     10% :  -2.568413277899264
```

Рисинок 3.13 – Результати тесту для даних S&P500 (після різниціювання)

З рисунку видно, що графік став більш стаціонарним, графіки ACF та PACF перестали показувати великі значення та тест показав значення P_Value набагато менше за 0,05, тому з таким графіком уже можна працювати. Тест Дікі-Фуллера показав результат менший 0,05, тож на цьому можна зупинитись і почати підбір параметрів p та q.

Далі за допомогою ACF та PACF потрібно обрати параметри p та q ,

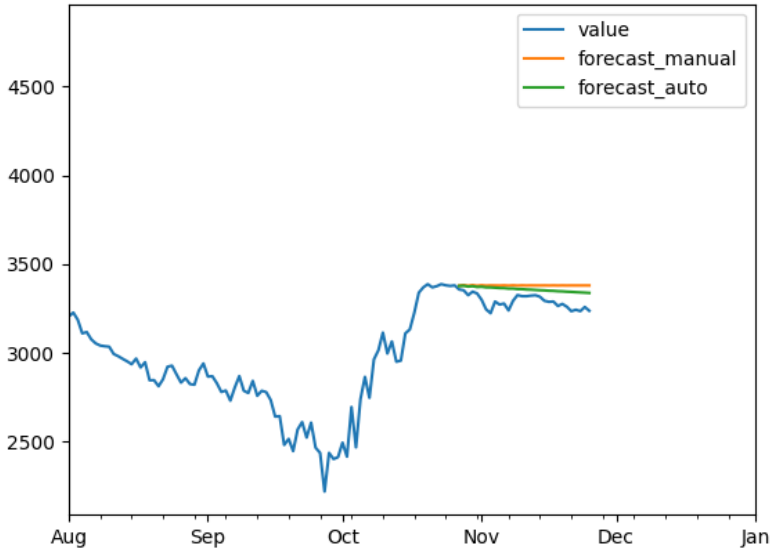
Оскільки графіки не показують плавного спаду та виразних сплесків на лагах p та q, то виходячи з інформації, що отримав з графіків я маю самостійно визначити параметри, таким чином для моделі підібраної вручну я обрав параметри порядку ARIMA(p,d,q)=ARIMA(2,1,2).

Підбір параметрів для руху курсу цін на золото є аналогічним, тому наводжу уже фінальні результати підбору для золота.

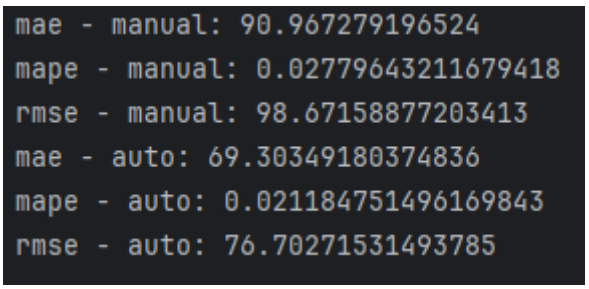
Обраний мною вручну набір параметрів для прогнозування – парметри порядку моделі ARIMA(p,d,q)=ARIMA(0,2,1)

3.5 Прогнозування за допомогою ARIMA

Прогнозуємо рух індексу S&P 500, у ході побудови моделі використовув два види набору параметрів, перший це підібрані самостійно параметри порядку ARIMA(p,d,q)=ARIMA(2,1,2), другий набір це підібрані параметри методом мінімізації показників AIC та SIC. Отримали результати прогнозування разом із показниками точності:



Рисинок 3.14 – Графік результатів прогнозування руху S&P 500 (ARIMA)



Рисинок 3.15 – Оцінка якості результатів прогнозування руху S&P 500 за встановленими метриками (ARIMA)

При прогнозуванні довжиною у один місяць з використанням ручного підбору параметрів (manual):

1) MAE – 90.967279196524;

2) MAPE – 0.02779643211679418;

3) RMSE – 98.67158877203413.

При прогнозуванні довжиною у один місяць з використанням автоматичного підбору параметрів (auto):

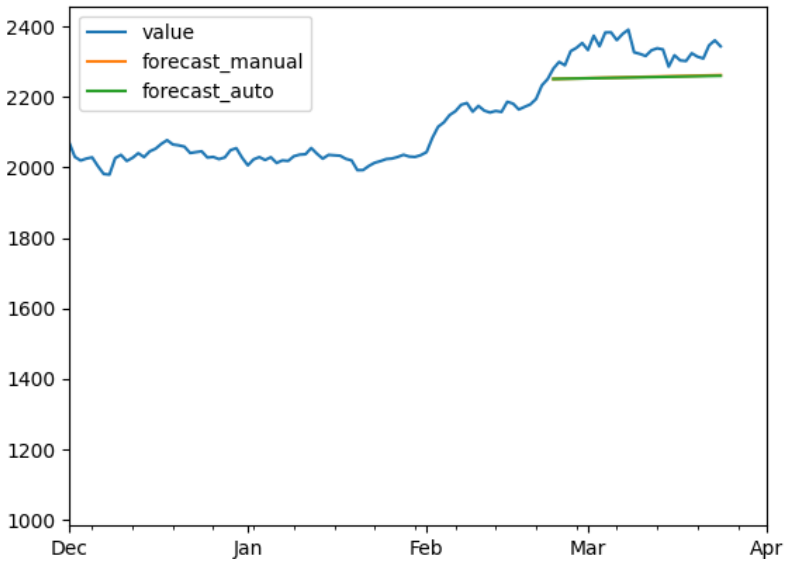
1) MAE – 69.30349180374836;

2) MAPE – 0.021184751496169843;

3) RMSE – 76.70271531493785.

З результатів можна зробити висновок, що в залежності від задач які перед нами стоять та досвіду аналітика можна будувати кращі моделі, ніж підібрані за мінімізацією критеріїв. Але в даному прикладі автоматиний підбір краще вловив тренд руху даних.

Прогнозуємо рух курсу золота, у ході побудови моделі використовували два види набору параметрів, перший це підібрані самостійно параметри порядку моделі ARIMA(p,d,q)=ARIMA(0,2,1), другий набір це підібрані параметри методом мінімізації показників AIC та SIC, результати прогнозування разом із показниками точності:



Рисинок 3.16 – Графік результатів прогнозування курсу золота (ARIMA)

```
mae - manual: 77.3850508819979
mape - manual: 0.033008413316499556
rmse - manual: 82.90039457857189
mae - auto: 78.32818781497608
mape - auto: 0.03341276529206803
rmse - auto: 83.75434714070562
```

Рисинок 3.17 – Оцінка якості результатів прогнозування курсу золота за встановленими метриками (ARIMA)

При прогнозуванні довжиною у один місяць з використанням ручного підбору параметрів (manual):

1) MAE – 77.3850508819979;

2) MAPE – 0.033008413316499556;

3) RMSE – 82.90039457857189.

При прогнозуванні довжиною у один місяць з використанням автоматичного підбору параметрів (auto):

1) MAE – 78.32818781497608;

2) MAPE – 0.03341276529206803;

3) RMSE – 83.75434714070562.

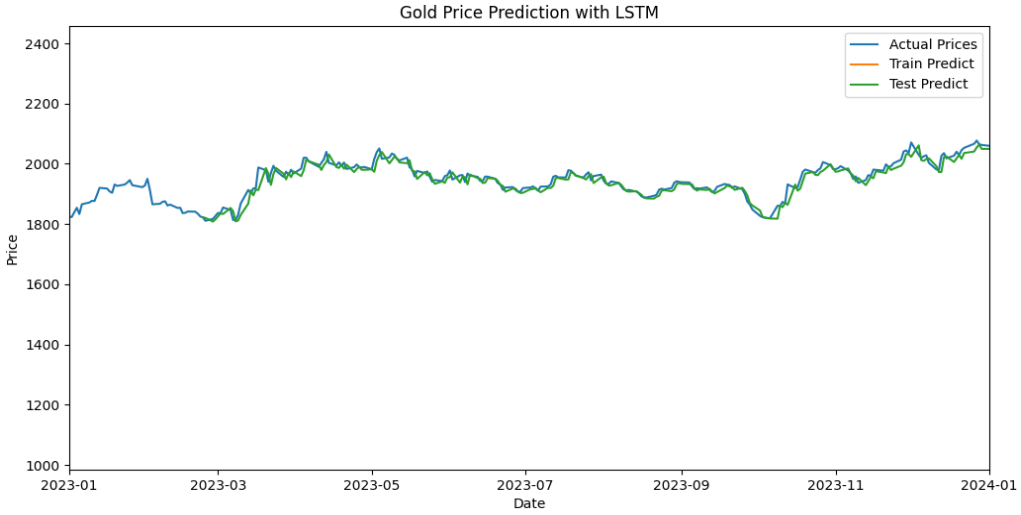
Результати показують, що ручний підбір параметрів перевершує автоматичний підбір за всіма трьома метриками. Це підтверджує, що аналітик з достатнім досвідом та знанням може налаштувати модель більш точно, враховуючи специфіку даних. Проте, різниця у результатах між ручним та автоматичним підбором є мінімальною.

Виходячи з досвіду роботи з моделлю, я рекомендую використовувати обидва підходи для досягнення найкращих результатів, особливо у випадках, коли важлива максимальна точність прогнозів. Це дозволяє уникнути упереджень та забезпечити врахування всіх важливих факторів, які можуть впливати на точність прогнозів.

3.6 Прогнозування за допомогою LSTM

Модель LSTM була побудована з двома шарами LSTM та вихідним шаром Dense. Це дозволило моделі враховувати складні залежності в даних. Я навчав модель протягом 100 епох з оптимізатором Adam і функцією втрат mean_squared_error. Це допомогло створити модель, яка на основі середнього квадратичного відхилення навчилась прогнозуванню даних.

Прогнозуємо рух курсу золота за допомогою моделі LSTM:



Рисинок 3.18 – Графік результатів прогнозування руху курсу золота (LSTM)

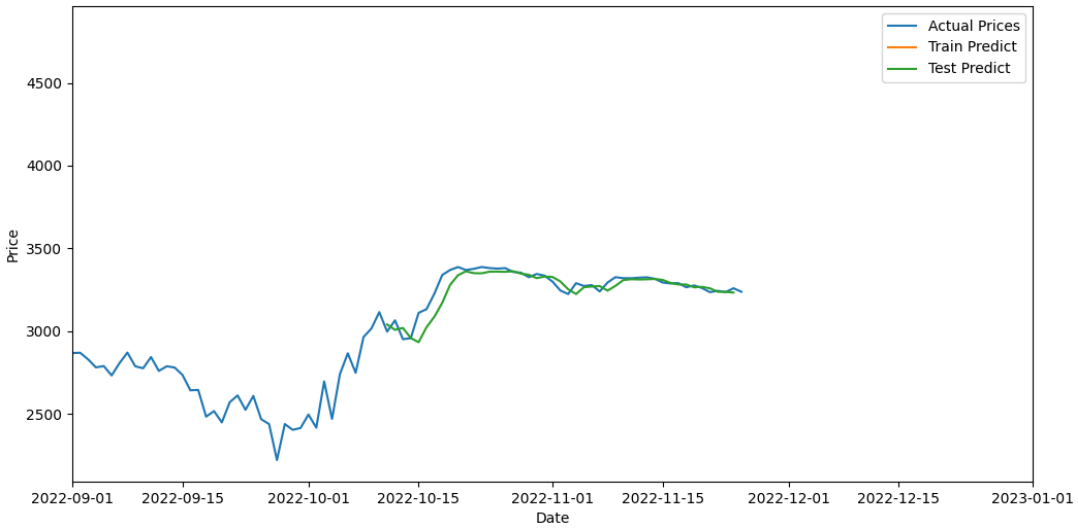
```
mae : 17.62069686813598
mape : 0.00847839903157342
rmse : 24.380956930363283
```

Рисинок 3.19 – Оцінка якості результатів прогнозування курсу золота за встановленими метриками (LSTM)

При прогнозуванні довжиною у один місяць отримали результати:

1) MAE – 17.6206986813598;
2) MAPE – 0.00847839903157342;
3) RMSE – 24.380956930363283.

Прогнозуємо рух індексу S&P 500 за допомогою моделі LSTM:



Рисинок 3.20 – Графік результатів прогнозування руху індексу S&P 500(LSTM)

```
mae : 35.16004752604168
mape : 0.01086407051149407
rmse : 54.52000781099759
```

Рисинок 3.21 – Оцінка якості результатів прогнозування руху індексу S&P 500 за встановленими метриками (LSTM)

При прогнозуванні довжиною у один місяць отримали результати:

1) MAE – 35.16004752604168;

2) MAPE – 0.01086407051149407;

3) RMSE – 54.52000781099759.

Результати прогнозування та оцінка їх точності вказують на те що результати дуже точні та майже повторюють рух реальних даних, добре враховуючи сеонність та тренд руху.

3.7 Порівняння методів прогнозування

Я провів дослідження та отримав наступні результати з прогнозування індексу S&P 500. Модель LSTM показала високу точність прогнозування з меншими значеннями середньої абсолютної похибки (MAE) та кореня середнього квадратичного відхилення (RMSE) порівняно з моделлю ARIMA:

Таблиця 1. Порівняння результатів прогнозування для індекса S&P 500

Результати	Arima вручну	Arima автоматична	LSTM
MAPE	2.7%	2.1%	1%
MAE	90.96	69.30	35.16
RMSE	98.6	76.7	54.52
Проміжок прогнозування	1 місяць	1 місяць	1 місяць

Я провів дослідження та отримав наступні результати з прогнозування курсу золота . Модель LSTM показала високу точність прогнозування з меншими значеннями середньої абсолютної похибки (MAE) та кореня середнього квадратичного відхилення (RMSE) порівняно з моделлю ARIMA:

Таблиця 2. Порівняння результатів прогнозування для курсу золота

Результати	Arima вручну	Arima автоматична	LSTM
MAPE	3,3%	3,3%	0,8%
MAE	77,38	78,32	17,62
RMSE	82,9	83,75	24,38
Проміжок прогнозування	1 місяць	1 місяць	1 місяць

Модель LSTM демонструє найкращі результати для обох видів даних, з найнижчими показниками похибок прогнозування (MAE, MAPE, RMSE). Це свідчить про її здатність ефективно обробляти часові ряди та враховувати складні залежності у даних. LSTM є особливо корисною для задач з довготривалими залежностями та складними патернами.

Автоматичний підбір параметрів ARIMA показує кращі результати у порівнянні з ручним підбором для індексу S&P 500, але трохи гірші для цін на золото. Це свідчить про те, що автоматизація процесу підбору параметрів може бути корисною, але іноді може поступатися ручному підбору, залежно від специфіки даних.

Ручний підбір параметрів ARIMA показує гірші результати у порівнянні з LSTM для обох типів даних та поступається автоматичному підбору для індексу S&P 500. Однак, він все ще може бути корисним для аналітиків, які мають глибокі знання про дані та можуть вручну налаштувати модель для досягнення кращих результатів.

Також варто зазначити, що я обрав проміжок прогнозування саме у місяць, аби перевірити ефективність моделей прогнозування на великому проміжку часу,

адже саме такий проміжок часу є важливим для прийняття рішення щодо інвестування з метою заробітку або збереження коштів.

Провівши прогнозування на менших проміжках, таких як тиждень та три дні, я отримав більш точні результати для моделі ARIMA, не гірші за прогнозування за допомогою моделі LSTM, але така довжина прогнозування не дає можливості використати цю інформацію для заробітку і тому не є ефективною.

Таком чином, з результатів можна зроюити висновок, що модель LSTM виявилася найефективнішою для прогнозування як індексу S&P 500, так і цін на золото. Вона забезпечує високу точність та найнижчі показники похибок. Автоматичний підбір параметрів ARIMA може бути корисним інструментом, але його ефективність може варіюватися в залежності від даних. Ручний підбір параметрів ARIMA може бути корисним у певних умовах, але зазвичай поступається автоматичним методам та моделям на основі глибинного навчання.

3.8 Використані технології та інструменти

Python є основною мовою програмування, використаною в даній роботі. Це високорівнева, інтерпретована мова програмування, відома своєю простотою та потужністю. Python широко використовується в наукових дослідженнях, зокрема в області обробки даних та машинного навчання, завдяки великій кількості бібліотек, які полегшують ці процеси. У роботі були використані такі основні модулі.

1. *Pandas* – є однією з найпопулярніших бібліотек для обробки та аналізу даних у Python. Вона забезпечує швидку та гнучку роботу з табличними даними. У нашому проекті Pandas використовувалась для зчитування, очищення та підготовки даних з файлів, а також для проведення базових операцій з даними;

2. *Matplotlib* – є бібліотекою для візуалізації даних в Python. Вона дозволяє створювати статичні, анімаційні та інтерактивні графіки. Ми використовували Matplotlib для візуалізації цін на золото, а також для відображення результатів сезонної декомпозиції та прогнозів;

3. *Statsmodels* – є бібліотекою для статистичного моделювання в Python. Вона забезпечує інструменти для оцінки статистичних моделей та проведення тестів. У нашій роботі ми використовували Statsmodels для побудови моделі ARIMA (AutoRegressive Integrated Moving Average), яка є одним з класичних методів аналізу часових рядів. Ця бібліотека також була використана для проведення тесту Дікі-Фуллера, що допоміг оцінити стаціонарність часового ряду;

4. *Numpy* – є основною бібліотекою для роботи з масивами та числовими операціями в Python. Вона забезпечує високу продуктивність обчислень та багатий набір функцій для роботи з багатовимірними масивами. Ми використовували Numpy для різних числових операцій та обробки даних;

5. *Scikit-learn* – є популярною бібліотекою для машинного навчання в Python. Вона забезпечує великий набір інструментів для попередньої обробки даних, класифікації, регресії, кластеризації та оцінки моделей. У нашому проекті Scikit-learn використовувалась для масштабування даних за допомогою MinMaxScaler та для обчислення метрик точності прогнозів;

6. *TensorFlow* – є однією з найбільш популярних платформ для машинного навчання, розробленою компанією Google. Вона забезпечує гнучкий та ефективний інструментарій для побудови та навчання моделей машинного навчання. У нашому проекті TensorFlow використовувалася для створення та навчання моделі LSTM (Long Short-Term Memory);

7. *Keras* – є високорівневою нейронною мережею API, що працює на вершині TensorFlow. Вона забезпечує зручний та зрозумілий інтерфейс для побудови та навчання моделей глибокого навчання. Ми використовували Keras для створення багатошарової моделі LSTM, що включала дві LSTM шари та Dense шар для прогнозування цін на золото.

У підсумку, для роботи з реалізації моделей прогнозування цін на золото я використовував широкий набір сучасних інструментів та бібліотек. Це дозволило ефективно обробляти дані, проводити статистичний аналіз та будувати точні моделі машинного навчання. Комбінація класичних методів, таких як ARIMA, з сучасними нейронними мережами, такими як LSTM, забезпечила високу точність прогнозування та надійність отриманих результатів.

3.9 Висновок до розділу 3

Цей розділ дослідження представляє глибокий аналіз і порівняння двох популярних методів прогнозування фінансових часових рядів – ARIMA та LSTM. Обравши в якості об'єктів дослідження фондовий індекс S&P 500 та курс золота, я зробив вагомий внесок у розуміння цих методів та їх ефективності в контексті фінансових даних.

Процес дослідження розпочався зі збору та обробки даних, що є фундаментальним етапом для будь-якого аналітичного дослідження. Очищення даних від аномалій, заповнення пропусків і нормалізація значень дозволили підготувати високоякісні вхідні дані для моделей. Підготовка також містить використання сезонної декомпозиції, яка допомогла виділити основні компоненти часових рядів, такі як – тренд, сезонність і випадкові коливання. Цей підхід забезпечив детальне та суттєве розуміння структури даних та допоміг ідентифікувати ключові фактори, що впливають на прогнозування.

Модель ARIMA, з її здатністю моделювати лінійні залежності та стаціонарні ряди, продемонструвала свою надійність. Використовуючи автокореляційну функцію (ACF) та часткову автокореляційну функцію (PACF), я визначив оптимальні параметри моделі, що дозволило досягти високої точності прогнозів. Тест Дікі-Фуллера став важливим інструментом для перевірки стаціонарності даних, а критерії AIC та BIC допомогли оптимізувати модель.

LSTM, як передовий метод машинного навчання, показав свою ефективність у роботі з довгостроковими залежностями у фінансових даних. Архітектура LSTM з її вхідними, забувальними та вихідними воротами дозволила моделі ефективно враховувати складні патерни у часових рядах. Тренування моделі на даних про індекс S&P 500 та курс золота підтвердило її здатність прогнозувати з високою точністю.

Порівняння результатів прогнозування між методами ARIMA та LSTM чітко продемострувало сильні та слабкі сторони кожної з них. Модель LSTM заебзпечує найкращі результати з найнижчими показниками похибок (MAE, MAPE, RMSE), що свідчить про її здатність ефективно обробляти часові ряди та враховувати складні залежності в даних. Автоматичний підбір параметрів ARIMA показав кращі результати у порівнянні з ручним підбором для індексу S&P 500, але трохи гірші для цін на золото, що свідчить про необхідність поєднання автоматичних і ручних методів для досягнення найкращих результатів а також про можливість покращення результатів за допомогою залучення додаткового аналізу та досвіду.

Завершуючи цей розділ, можна зробити висновок, що модель LSTM виявилася хоч і повільнішою та важчою у виконанні, проте найефективнішою для прогнозування як індексу S&P 500, так і цін на золото. Вона забезпечує високу точність та найнижчі показники похибок, що робить її надзвичайно корисною для фінансових аналітиків, інвесторів та дослідників. Модель ARIMA також є надійним інструментом, особливо коли параметри підбираються з урахуванням специфіки даних. Комбінація цих двох методів може забезпечити ще більш точні та надійні прогнози, враховуючи всі важливі фактори, що впливають на фінансові ринки.

Таким чином, це дослідження не лише поглиблює розуміння методів ARIMA та LSTM для прогнозування сучасних нестійких фінансових даних, але й надає цінні рекомендації для їх ефективного застосування в реальних умовах, що може значно підвищити точність прогнозів та прийняття рішень у фінансовій сфері.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У цьому розділі буде проведено оцінку основних характеристик майбутнього програмного продукту для розпізнавання емоцій на людському обличчі. Це дослідження сприятиме проведенню всіх необхідних аналізів, що дозволить якісно вивчити це питання не лише в Україні, а й у всьому світі.

У цьому дослідженні розглянуто різні варіанти реалізації для забезпечення найбільш коректної та оптимальної стратегії вибору, яка впливає на економічні фактори та сумісність із майбутнім програмним продуктом. Для цього застосовувався апарат функціонально-вартісного аналізу (ФВА).

Функціонально-вартісний аналіз є технологією, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва та кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих годин, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

4.1 Постановка задачі проектування

У роботі використовується метод функціонально-вартісного аналізу (ФВА) для проведення техніко-економічного аналізу розробки системи розпізнавання емоцій на людському обличчі. Оскільки рішення щодо проектування та реалізації

компонентів впливають на всю систему, кожна окрема підсистема повинна відповідати вимогам системи в цілому. Таким чином, фактичний аналіз включає аналіз функцій програмного продукту, призначеного для збору, обробки та аналізу даних компанії.

Технічні вимоги до програмного продукту включають наступні пункти:

1) програмний продукт повинен функціонувати на персональних комп'ютерах зі стандартним набором компонентів;

2) інтерфейс повинен бути зручним та зрозумілим для користувача;

3) програмний продукт повинен забезпечувати швидку обробку даних та доступ до інформації в реальному часі;

4) програмний продукт повинен мати можливість зручного масштабування та обслуговування;

5) витрати на впровадження програмного продукту повинні бути мінімальними.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка можливого програмного продукту, яка дозволяє аналізувати різні характеристики, що безпосередньо впливають на стійкість підприємства. Беручи за основу цю функцію, можна виділити наступні:

1) F_1 – вибір мови програмування;
2) F_2 – вибір способу реалізації алгоритмів;
3) F_3 – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації:

1) Функція F_1:

а) Python.

б) C/C++.

2) Функція F_2:

а) використання готових бібліотек;

б) реалізація алгоритмів вручну.

3) Функція F_3:

а) середовище розробки – Jupyter Notebook;

б) середовище розробки – Pycharm IDE/VS Code.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рисунок 4.1).

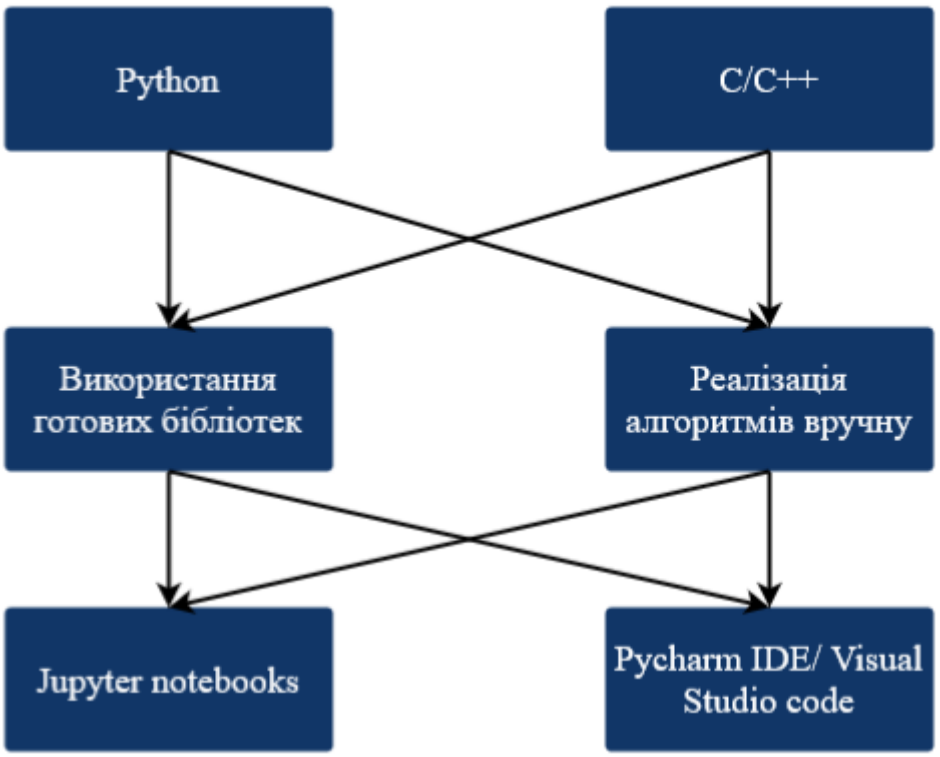


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіантів основних функцій. Позитивно-негативна матриця показана в таблиці 4.1.

Таблиця 4.1 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	А	Простота, доступність готових бібліотек	Повільність
	Б	Час виконання програми	Складність, потребує Більше часу для написання програми
F2	А	Доступність та легкість при написанні	Менша гнучкість
	Б	Реалізація функціоналу, який потрібен для програми	Додатковий час на реалізацію, можливі помилки
F3	А	Для кожної клітини можна обрати нову мову програмування	Відсутність дебагу коду
	Б	Багато інструментів, інтеграція з іншими сервісами	Підтримує лише одну мову програмування

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

1. Функція F_1 – перевагу даємо загальнодоступності. Для спрощення роботи по написанню коду варіант Б має бути відкинутий.

2. Функція F_2 – готові бібліотеки для мови програмування Python є

зручними та оптимізованими, тому варіант Б має бути відкинутий.

3. Функція F_3 – Обидва варіанти можна використати у розробці.

Таким чином, будемо розглядати такий варіанти реалізації ПП:

1) $F_{1а} – F_{2а} – F_{3а}$;

2) $F_{1а} – F_{2а} – F_{3б}$.

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

1) *X1* – швидкодія мови програмування;

2) *X2* – об’єм пам’яті для обчислень та збереження даних;

3) *X3* – час навчання даних;

4) *X4* – потенційний об’єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у таблиці 4.2.

Таблиця 4.2 – Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	85	110	160
Об’єм пам’яті	X2	Мб	100	80	50
Час попередньої обробки даних	X3	мс	120	85	60
Потенційний об’єм програмного коду	X4	кількість рядків коду	1000	650	500

За даними таблиці 4.2 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

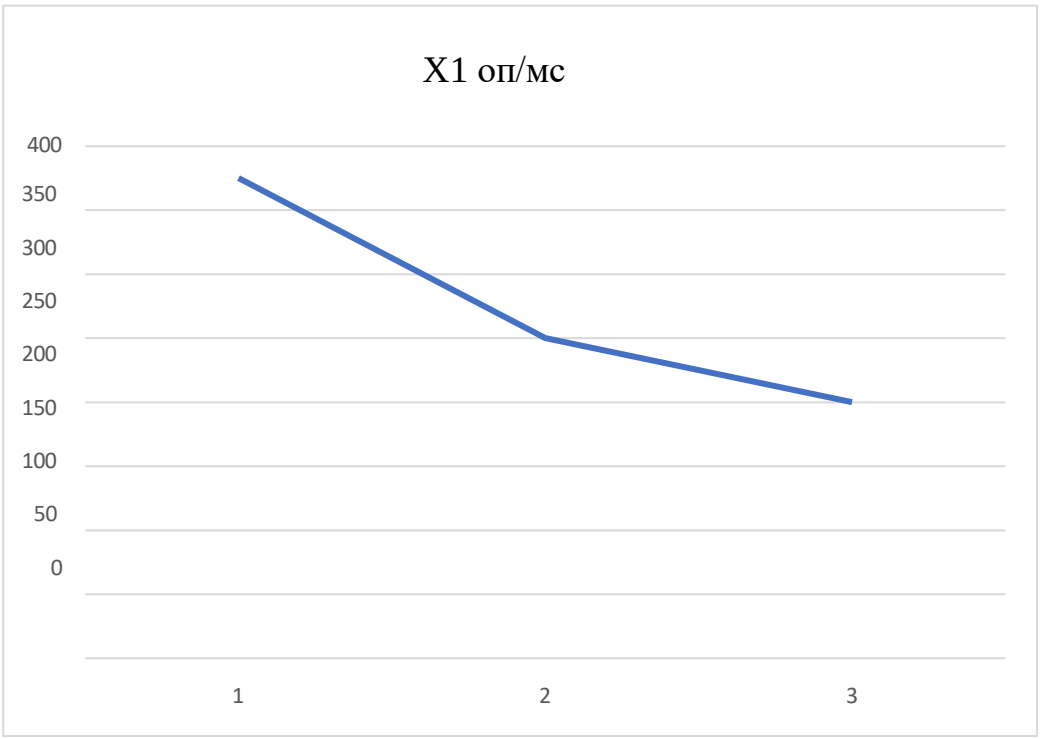


Рисунок 4.2 – X1, швидкодія мови програмування

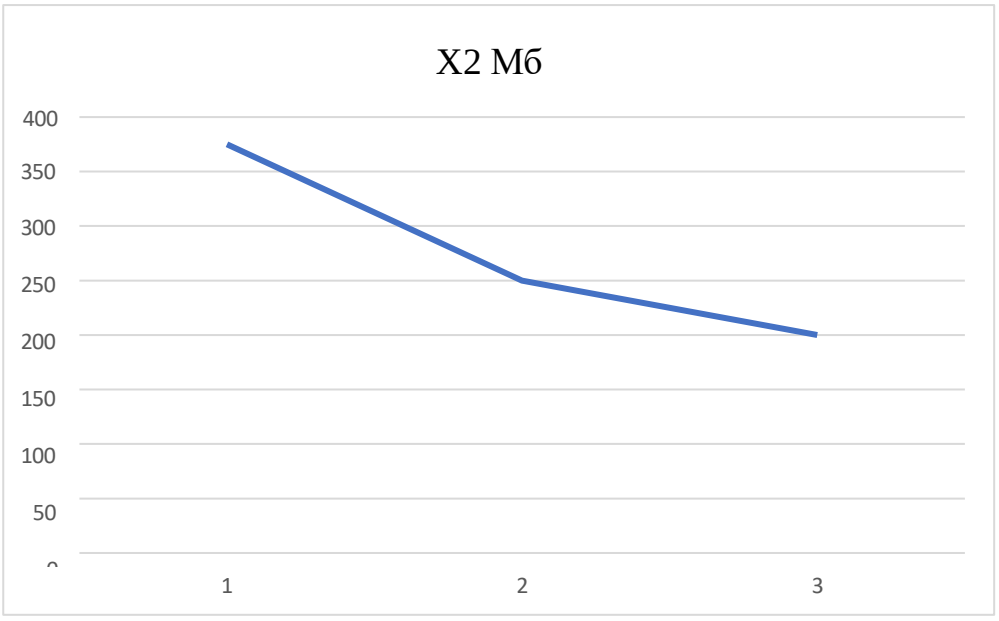


Рисунок 4.3 – X2, об’єм пам’яті

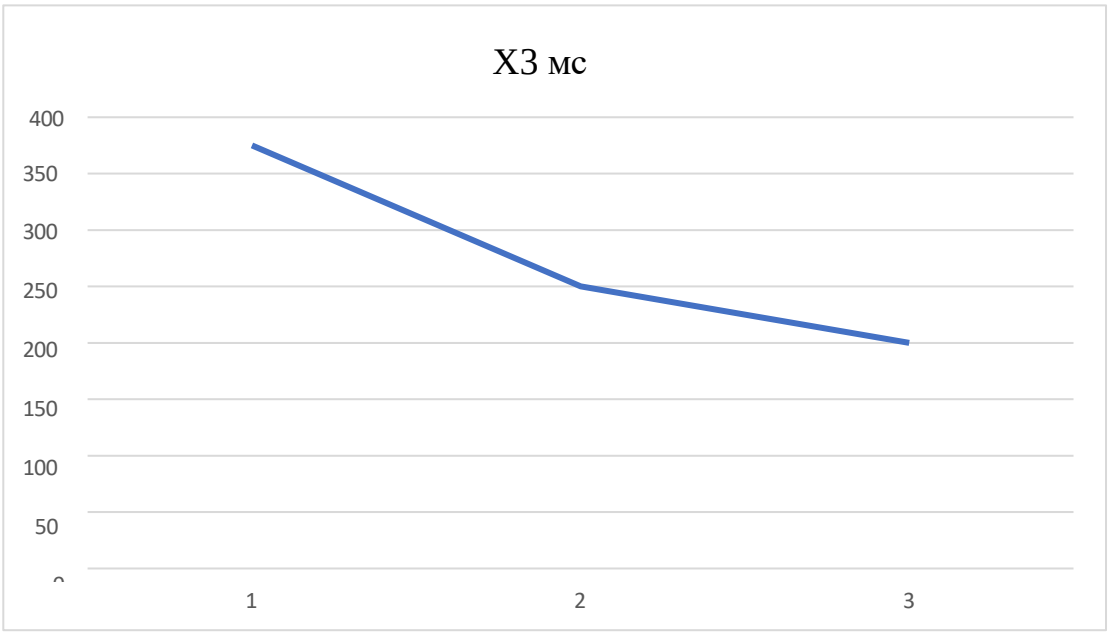


Рисунок 4.4 – X3, час попередньої обробки даних

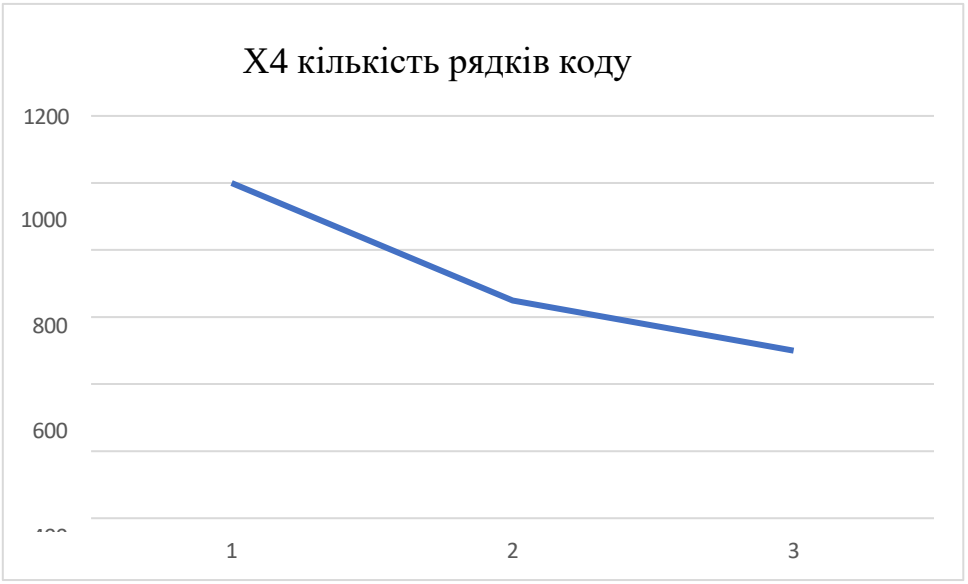


Рисунок 4.5 – X4, потенційний об’єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі

розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

1) визначення рівня значимості параметра присвоєнням різних рангів;

2) перевірку придатності експертних оцінок для подальшого використання;

3) визначення оцінки попарного пріоритету параметрів;

4) обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати ранжування пераметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг парметра за оцінкою експертів						Сума рангів R_i	Відхилення Δ_i	Δ_{i2}
			1	2	3	4	5	6			
X1	Швидкодія мови програмування	Оп/мс	2	2	1	1	1	2	9	-6	36
X2	Об’єм пам’яті	Mb	3	3	5	5	4	4	24	9	81
X3	Час попередньої обробки даних	мс	3	1	1	2	2	2	11	-4	16
X4	Потенційний об’єм програмного коду	Кількість рядків коду	2	4	3	2	3	2	16	1	1
	Разом		10	10	10	10	10	10	60		134

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

1) сума рангів кожного з параметрів і загальна сума рангів;

$$R_i = \sum_{j=1}^{N} r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 60 \quad (4.1)$$

де N – число експертів;

n – кількість параметрів.

2) середня сума рангів;

$$T = \frac{1}{n} R_{ij} = 15 \quad (4.2)$$

3) відхилення суми рангів кожного параметра від середньої суми рангів;

$$\Delta_i = R_i - T \quad (4.3)$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

4) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^{N} \Delta_i^2 = 134 \quad (4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 134}{6^2(4^3 - 4)} = 0{,}744 > W_k = 0{,}67 \quad (4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів.

Параметри	Експерти						Кінцева оцінка	Числове значення
	1	2	3	4	5	6		
X1 і X2	<	<	<	<	<	<	<	0.5
X1 і X3	<	>	=	<	<	=	<	0.5
X1 і X4	=	<	<	<	<	=	<	0.5
X2 і X3	=	>	>	>	>	>	>	1.5
X2 і X4	>	<	>	>	>	>	>	1.5
X3 і X4	>	<	<	=	<	=	<	0.5

Числове значення, що визначає ступінь переваги i-го параметра над j-тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases} \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю $\mathrm{A} = \|a_{ij}\|$.

Для кожного параметра зробимо розрахунок вагомості $K_{6\mathrm{i}}$ за наступними формулами:

$$K_{\mathrm{Bi}} = \frac{b_i}{\sum_{i=1}^{n} b_i} \quad (4.7)$$

$$b_i = \sum_{i=1}^{N} a_{ij} \quad (4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\mathrm{Bi}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'} \quad (4.9)$$

$$b_i' = \sum_{i=1}^{N} a_{ij} b_j \quad (4.10)$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер.	
	X1	X2	X3	X4	b_i	$К_{вi}$	$b1_i$	$К1_{вi}$	$b2_i$	$К2_{вi}$
X1	1	0,5	0,5	0,5	2,5	0,16	9,5	0,16	34,125	0,16
X2	1,5	1	1,5	1,5	5,5	0,34	21,25	0,35	77,875	0,36
X3	1,5	0,5	1	0,5	3,5	0,22	12,25	0,21	41,875	0,2
X4	1,5	0,5	1,5	1	4,5	0,28	16,25	0,28	59,125	0,28
Всього:					16	1	59	1	213	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів) X1 (швидкість роботи мови програми) , X2 (Об'єм пам'яті) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра $X4$ (потенційний об'єм програмного коду) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K(j) = \sum_{i=1}^{n} K_{6i,j} B_{i,j} \qquad (4.11)$$

де n – кількість параметрів;

K_{si} – коефіцієнт вагомості i-го параметра;

B_i – оцінка i-го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант релізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X1	110	5	0,16	0,8
F2	A	X2	80	3	0,36	1,08
F3	A	X4	800	7	0,2	1,4
	F	X4	600	4	0,2	0,8

За даними з таблиці 4.6 за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \cdots + K_{\text{ТУ}}[F_{zk}] \quad (4.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 0.8 + 1.08 + 1.4 = 3.28$$

$$K_{K2} = 0.8 + 1.08 + 0.8 = 2.68$$

Як видно з розрахунків, кращим є 1 варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1) розробка проекту програмного продукту;
2) розробка програмної оболонки.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$\mathrm{T_O} = \mathrm{T_P} \cdot \mathrm{K_П} \cdot \mathrm{K_{СК}} \cdot \mathrm{K_М} \cdot \mathrm{K_{СТ}} \cdot \mathrm{K_{СТ.М}} \qquad (4.13)$$

де $\mathrm{T_P}$ – трудомісткість розробки ПП;

$\mathrm{K_П}$ – поправочний коефіцієнт;

$\mathrm{K_{СК}}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$\mathrm{K_{Ст}}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$\mathrm{K_{Ст.м}}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $\mathrm{T_P} = 90$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання, $K_{\text{п}} = 1{,}8$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: К використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $\mathrm{K_{СТ}} = 0.9$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$\mathrm{T_1} = 90 \cdot 1.8 \cdot 0.9 = 145{,}8 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $\mathrm{T_P} = 29$ людино-днів, $\mathrm{K} = 0.9, \mathrm{K_{СК}} = 1$, $K_{\mathrm{СТ}} = 0.8$:

$$\mathrm{T_2} = 29 \cdot 0.9 \cdot 0.8 = 20.88 \text{ людино-днів.}$$

Для завдання три (А) використовується алгоритм третьої групи складності, степінь новизни Г тобто

$$T_р = 8 \text{ людино-днів}, К_{СК} = 1,\ К_{СТ} = 0.6$$
$$T_{З(а)} = 8 \cdot 0.6 = 4.8 \text{ людино-днів}$$

Для завдання три (Б) використовується алгоритм другої групи складності, степінь новизни Г, тобто

$$T_р = 12 \text{ людино-днів},\ К_{СК} = 1,\ К_{СТ} = 0.6,\ T_{З(б)} = 12 \cdot 0.6 = 7.2 \text{ людино-днів}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (145{,}8 + 20.88 + 4.8) \cdot 8 = 1371{,}84 \text{ людино-годин.}$$
$$T_{II} = (145{,}8 + 20.88 + 7.2) \cdot 8 = 1391{,}04 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 22300 грн., один аналітик в області даних з окладом 20000. Визначимо середню зарплату за годину за формулою:

$$С_Ч = \frac{М}{T_m \cdot t} \text{ грН} \quad (4.14)$$

де М – місячний оклад працівників;

T_m – кількість робочих днів тиждень;

t – кількість робочих годин в день.

$$С_Ч = \frac{22300 + 22300 + 20000}{3 \cdot 21 \cdot 8} = 128{,}17 \text{ грн.} \quad (4.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$С_{ЗП} = С_ч \cdot T_i \cdot К_д, \quad (4.16)$$

де $С_ч$ – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

Кд – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

1) $С_{ЗП} = 128{,}17 \cdot 1371{,}84 \cdot 1.2 = 210994{,}48$ грн.

2) $С_{ЗП} = 128{,}17 \cdot 1391{,}04 \cdot 1.2 = 213947{,}52$ грн.

Відрахування на єдиний соціальний внесок становить 22% :

1) $С_{вІд} = С_{ЗП} \cdot 0.22 = 210994,48 \cdot 0.22 = 46418,79$ грн.

2) $С_{вІд} = С_{ЗП} \cdot 0.22 = 213947,52 \cdot 0.22 = 47068,45$ грн.

Тепер визначимо витрати на оплату однієї машино-години. ($С_М$)

Так як одна ЕОМ обслуговує одного програміста з окладом 22300 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$С_Г = 12 \cdot М \cdot К_З = 12 \cdot 22300 \cdot 0,2 = 53520 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$С_{ЗП} = C_Г \cdot (1 + K_З) = 53520 \cdot (1 + 0.2) = 64224 \text{ грн.}$$

Відрахування на соціальний внесок:

$$С_{ВІД} = С_{ЗП} \cdot 0.22 = 64224 \cdot 0,22 = 14129,28 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 50000 грн.

$$С_А = К_{ТМ} \cdot К_А \cdot Ц_{ПР} = 1.4 \cdot 0.25 \cdot 50000 = 17500 \text{ грн.},$$

де $К_{ТМ}$ - коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$К_А$ – річна норма амортизації;

ЦпР – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$С_P = К_{ТМ} \cdot Ц_{ПР} \cdot К_P = 1.4 \cdot 50000 \cdot 0.08 = 5600 \text{ грн.},$$

де $К_P$ - відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$Т_{ЕФ} = (Д_К - Д_В - Д_С - Д_Р) \cdot t_З \cdot К_В = (365 - 104 - 12 - 16) \cdot 8 \cdot 0.35 = 627,2 \text{ години}$$

де Дк – календарна кількість днів у році;

Дв, Дс – відповідно кількість вихідних та святкових днів;

$Д_Р$ – кількість днів планових ремонтів устаткування;

t-кількість робочих годин в день;

$К_В$ – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$С_{ЕЛ} = Т_{ЕФ} \cdot N_С \cdot К_З \cdot Ц_{ЕН} = 627,2 \cdot 0,2 \cdot 0,3 \cdot 5,23 = 196,81 \text{ грн.},$$

де N_C – середньо-споживча потужність приладу;

$K_З$ – коефіцієнтом зайнятості приладу;

ЦЕН – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_Н = \text{ЦпР}\ ^0 0.67 = 50000 \cdot 0{,}67 = 33500 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{ЕКС} = C_{ЗП} + C_{ВІД} + C_А + C_Р + C_{ЕЛ} + C_Н, \quad (4.17)$$

$$C_{ЕКС} = 64224 + 14129{,}28 + 17500 + 5600 + 183{,}27 + 33500 = 135136{,}55 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{М-Г} = C_{ЕКС}/T_{ЕФ} = 135136{,}55/627{,}2 = 215{,}46 \text{ грн / год.}$$

Оскільки в даному випадку всі роботи, які пов‘язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_М = C_{М-Г} \cdot T \quad (4.18)$$

1) $C_М = 215{,}46 \cdot 1371{,}84 = 295576{,}65$ грн.

2) $C_M = 215{,}46 \cdot 1391{,}04 = 299713{,}48$ грн.

Накладні витрати складають 67% від заробітної плати:

$$C_Н = C_{ЗП} \cdot 0{,}67 \quad (4.19)$$

1) $C_Н = 295576{,}65 \cdot 0{,}67 = 198036{,}36$ грн.

2) $C_Н = 299713{,}48 \cdot 0{,}67 = 200808{,}03$ грн.

Отже, вартість розробки ПП за варіантами становить:

$$C_{пп} = C_{ЗП} + C_{ВІд} + C_М + C_Н \quad (4.20)$$

1) $C_{Пп} = 210994{,}48 + 46418{,}79 + 295576{,}65 + 198036{,}36 = 751026{,}28$ грн.

2) $C_{ПП} = 213947{,}52 + 47068{,}45 + 299713{,}48 + 200808{,}03 = 761537{,}48$ грн.

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$\mathrm{K_{TEP}}_j = \mathrm{K_{Kj}}/\mathrm{C_{Фj}},$$
$$\mathrm{K_{TEP1}} = 3{,}28/751026{,}28 = 4{,}367 \cdot 10^{-6} \quad (4.21)$$
$$\mathrm{K_{TEP2}} = 2{,}68/761537{,}48 = 3{,}519 \cdot 10^{-6}$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $\mathrm{K_{TEP1}} = 4{,}367 \cdot 10^{-6}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості.

$K_{\mathrm{TEP}} = 4{,}367 \cdot 10^{-6}$.

Цей варіант реалізації програмного продукту має такі параметри:

1) Вибір мови програмування – Python;
2) Використання готових бібліотек;
3) Середовище розробки – Jupyter Notebook.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, швидку реалізацію програми та доступний функціонал для роботи.

4.8 Висновки до четвертого розділу

В даній частині було проведено повний функціонально-вартісний аналіз програмного продукту. Також було знайдено оцінку основних функцій програмного продукту.

В результаті виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, було визначено та проведено оцінку основних функцій програмного продукту, а також знайдено параметри, які його характеризують.

На основі аналізу вибрано варіант реалізації програмного продукту.

ВИСНОВКИ

У цій дипломній роботі було проведено глибоке дослідження та порівняння двох популярних методів прогнозування фінансових часових рядів – ARIMA та LSTM – з метою оцінки їх ефективності на прикладі фондового індексу S&P 500 та курсу золота. Розуміння цих методів та їх застосування є важливим для підвищення точності прогнозів, що має велике значення для аналітиків, інвесторів та дослідників.

Перший розділ надав фундаментальне розуміння теоретичних основ методів прогнозування часових рядів. Було розглянуто основи моделі ARIMA, яка поєднує авторегресію (AR), інтеграцію (I) та ковзне середнє (MA) для аналізу та прогнозування стаціонарних часових рядів та рядів що можуть бути приведеними до стаціонарного стану. Було описано, як обираються параметри моделі за допомогою автокореляційної функції (ACF) та часткової автокореляційної функції (PACF), а також як застосовуються критерії AIC та BIC для оцінки моделей. Окрім цього, було введено тест Дікі-Фуллера для перевірки стаціонарності даних, що є критично важливим для коректного застосування моделі ARIMA.

У цьому розділі також детально розглянуто основи роботи з LSTM – потужним методом машинного навчання для роботи з часовими рядами, що має здатність враховувати довгострокові залежності в даних завдяки своїй унікальній архітектурі, яка включає вхідні, забувальні та вихідні ворота.

Другий розділ був присвячений практичному застосуванню методів ARIMA та LSTM для прогнозування фондового індексу S&P 500 та курсу золота. Було проведено збір та обробку даних, включаючи очищення від аномалій, заповнення пропусків та нормалізацію значень. Використання сезонної декомпозиції допомогло виділити основні компоненти часових рядів, такі як тренд, сезонність та залишки, що значно покращило розуміння структури даних.

Для моделі ARIMA були обрані оптимальні параметри за допомогою ACF, PACF, AIC та BIC, а також було проведено тест Дікі-Фуллера для перевірки стаціонарності даних. Модель ARIMA продемонструвала свою ефективність при короткострокових прогнозах, особливо коли параметри були налаштовані вручну.

Модель LSTM була побудована з двома шарами LSTM та вихідним шаром Dense. Це дозволило моделі враховувати складні залежності в даних і забезпечити високу точність прогнозів для обох видів фінансових показників.

У третьому розділі було проведено детальне порівняння результатів прогнозування між методами ARIMA та LSTM. Модель LSTM продемонструвала найкращі результати для обох видів даних, з найнижчими показниками похибок прогнозування (MAE, MAPE, RMSE). Проте, метод ARIMA також показав свою актуальність, особливо для короткострокових прогнозів та в умовах, де потрібна швидка та надійна оцінка.

Ця робота показала, що модель LSTM є найефективнішою для прогнозування фінансових даних з метою вигідного використання коштів, а модель ARIMA, як більш старий та менш потужний метод, все ще залишається актуальною на коротких проміжках часу, особливо при залученні серйозних експертів з фінансової області, які можуть оптимально налаштувати параметри моделі вручну.

Таким чином, у дипломній роботі не лише поглибилине розуміння методів ARIMA та LSTM, але й надані цінні рекомендації для їх ефективного застосування в реальних умовах, що може значно підвищити точність прогнозів та прийняття рішень у фінансовій сфері. Комбінація класичних та сучасних методів прогнозування дозволяє отримати найбільш точні та надійні результати, враховуючи всі важливі фактори, що впливають на фінансові ринки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ali, F., Suri, P., Pandey, S., Kathuria, S., Kumar, A., & Negi, P. (2023, April). Prediction of Stock Market Analysis by Artificial Intelligence. In 2023 IEEE *International Conference on Contemporary Computing and Communications* (*InC4*) (Vol. 1, pp. 1-5). IEEE. DOI: 10.1109/InC457730.2023.10263023
2. Herman D, Googin C, Liu X, Galda A, Safro I, Sun Y, Pistoia M, Alex-eev Y (2022) A survey of quantum computing for finance. Papers2201.02773, arXiv.org.https://ideas.repec.org/p/arx/papers/2201.02773.html
3. Lesik, I. (2019). Socio-Economic Risks of Unemployment for the Agricultural Market Infrastructure of Ukraine. *Modern Economics*, 17(2019), 127-132. DOI: 10.31521/modecon.V17(2019)-20 [in Ukrainian].
4. Angelina, S., Nugraha, N. M. (2020). Effects of Monetary Policy on Inflation and National Economy Based on Analysis of Bank Indonesia Annual Report. *Technium Soc. Sci. J*., 10, 423.
5. Nyoni, T. Modeling and forecasting inflation in Kenya: Recent in-sights from ARIMA and GARCH analysis. Dimorian Review, 5(6), 16–40 (2018).
6. Maheshwari, R., Kapoor, V. Predicting the NSE stock index trends considering global financial variables and ARIMA model. Journal of Statistics and Management Systems. 25(7): 1513-1522 (2022). DOI: 10.1080/09720510.2022.2130563
7. Sasi, A., Subramanian, T. Comparative analysis of ARIMA and dou-ble exponential smoothing for forecasting rice sales in fair price shop. Journal of Statistics and Management Systems. 25(7): 1601-1619 (2022). DOI: 10.1080/09720510.2022.2130572.
8. Li, Ping, Qiang Wu, Christopher J. Burges. Mcrank: Learning to rank using multiple classification and gradient boosting: *Advances in Neural Information Processing Systems*, 2007. 120 p.
9. Ozbayoglu A.M., Gudelek G.U., and Omer Berat Sezer. Deep learning for financialapplications : A survey. *arXiv preprint arXiv:2002.05786*, 2020.

10. Sendhil M., Spiess J. Machine learning: An applied econometric approach. Journalof Economic Perspectives, 31(2):87–106, May 2017.
11. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Networks*, 61:85–117, 2015.
12. Wang X, Kang Y, Hyndman RJ, et al. Distributed ARIMA models for ultra-long time series. *arXiv e-prints*, arXiv:2007.09577, 2020.
13. Theano development team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, arXiv:1605.02688, 2016.\
14. Ji L, Zou Y, He K, et al. Carbon futures price forecasting based with ARIMA-CNN-LSTM model. *Procedia Computer Science*, 162:33–38, 2019.
15. Kim TY and Cho SB. Predicting residential energy consumption using CNN-LSTM neural networks. Energy, 182:72–81, 2019.

ДОДАТОК А (LSTM)

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error, mean_squared_error
file_path = 'gold_new.txt'
data = pd.read_csv(file_path, delimiter='\t', header=None, names=['Date', 'Price'])
data['Date'] = pd.to_datetime(data['Date'], format='%d.%m.%Y', errors='coerce')
data['Price'] = data['Price'].str.replace(',', '.').astype(float)
data.dropna(subset=['Date', 'Price'], inplace=True)
data.set_index('Date', inplace=True)
plt.figure(figsize=(12, 6))
plt.plot(data, label='Gold Price')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Gold Price (2010-2023)')
plt.legend()
plt.show()
result = seasonal_decompose(data['Price'], model='multiplicative', period=365)
result.plot()
plt.show()
trend = result.trend
seasonal = result.seasonal
residual = result.resid
print("Trend:")
print(trend.dropna().head())
print("\nSeasonal:")
print(seasonal.dropna().head())
print("\nResidual:")
print(residual.dropna().head())
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data['Price'].values.reshape(-1, 1))
train_size = int(len(scaled_data) * 0.9)
```

```
test_size = len(scaled_data) - train_size
train_data, test_data = scaled_data[0:train_size], scaled_data[train_size:len(scaled_data)]
def create_dataset(dataset, time_step=1):
    X, Y = [], []
    for i in range(len(dataset) - time_step - 1):
        a = dataset[i:(i + time_step), 0]
        X.append(a)
        Y.append(dataset[i + time_step, 0])
    return np.array(X), np.array(Y)
time_step = 60
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(time_step, 1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X_train, y_train, batch_size=30, epochs=100)
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)
train_predict = scaler.inverse_transform(train_predict)
test_predict = scaler.inverse_transform(test_predict)
y_train = scaler.inverse_transform(y_train.reshape(-1, 1))
y_test = scaler.inverse_transform(y_test.reshape(-1, 1))
plt.figure(figsize=(12, 6))
plt.plot(data.index, data['Price'], label='Actual Prices')
plt.plot(data.index[time_step:len(train_predict) + time_step], train_predict, label='Train Predict')
plt.plot(data.index[len(train_predict) + (time_step * 2) + 1:len(data) - 1], test_predict, label='Test Predict')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Gold Price Prediction with LSTM')
plt.xlim(pd.Timestamp('2023-01-01'), pd.Timestamp('2024-01-01'))
plt.legend()
plt.show()
mse = mean_squared_error(y_test, test_predict)
print(f'Mean Squared Error: {mse}')
mae = mean_absolute_error(y_test, test_predict)
mape = mean_absolute_percentage_error(y_test, test_predict)
```

```
rmse = np.sqrt(mean_squared_error(y_test, test_predict))
print(f'mae : {mae}')
print(f'mape : {mape}')
print(f'rmse : {rmse}')
```

ДОДАТОК Б (ARIMA)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pmdarima as pm
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from pmdarima import auto_arima
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error, mean_squared_error
def df_testing(dataset):
    dftesting = adfuller(dataset,autolag='AIC')
    print("1. ADF : ",dftesting[0])
    print("2. P_Value : ",dftesting[1])
    print("3. Num Of Lags : ",dftesting[2])
    print("4. Num Of Observ. used for ADF reges. and Crit. val. calc. : ",dftesting[3])
    print("5. Crit. val. : ")
    for key, val in dftesting[4].items():
        print('\t',key,": ",val)
data = []
with open('gold_for_arima .txt', 'r') as file:
    for line in file:
        try:
            number = float(line.replace(',', '.'))  # Заміна коми на крапку для перетворення на float
            data.append(number)
        except ValueError:
            pass
start_date='2013-01-01'
dates = pd.date_range(start=start_date, periods=len(data), freq='D')
df = pd.DataFrame(data, columns=['value'], index=dates)
df.index.name = 'DATE'
df = df.dropna()
plt.figure(figsize=(14,7))
plt.plot(df.index, df['value'], label='Actual values')
plt.legend()
plt.show()
forecast_len=30
```

```
df_train=df[:-forecast_len].copy()
df_test=df[-forecast_len:].copy()
acf_or=plot_acf(df_train)
pacf_or=plot_pacf(df_train)
plt.show()
adf_test = df_testing(df)
print(df_testing)
df_train_diff = df_train.diff().dropna()
df_train_diff.plot()
plt.show()
df_testing(df_train_diff)
acf_diff = plot_acf(df_train_diff)
pacf_diff = plot_pacf(df_train_diff)
plt.show()
df_diff_2 = df_train_diff.diff().dropna()
df_diff_2.plot()
plt.show()
df_testing(df_diff_2)
acf_or=plot_acf(df_diff_2)
pacf_or=plot_pacf(df_diff_2)
plt.show()
model = ARIMA(df_train, order=(0,2,1))
model_fit = model.fit()
print(model_fit.summary())
residuals = model_fit.resid[1:]
fig, ax = plt.subplots(1,2)
residuals.plot(title='Residuals', ax=ax[0])
residuals.plot(title='Density', kind='kde', ax=ax[1])
plt.show()
acf_res = plot_acf(residuals)
pacf_res = plot_pacf(residuals)
forecast_test = model_fit.forecast(len(df_test))
df['forecast_manual'] = [None]*len(df_train) + list(forecast_test)
df.plot()
plt.show()
auto_arima = pm.auto_arima(df_train, stepwise=False, seasonal=False)
print(auto_arima)
print(auto_arima.summary())
forecast_test_auto = auto_arima.predict(n_periods=len(df_test))
df['forecast_auto'] = [None]*len(df_train) + list(forecast_test_auto)
df.plot()
```

```
plt.xlim(pd.Timestamp('2022-12-01'), pd.Timestamp('2023-04-01'))
plt.show()
mae = mean_absolute_error(df_test, forecast_test)
mape = mean_absolute_percentage_error(df_test, forecast_test)
rmse = np.sqrt(mean_squared_error(df_test, forecast_test))
print(f'mae - manual: {mae}')
print(f'mape - manual: {mape}')
print(f'rmse - manual: {rmse}')
mae = mean_absolute_error(df_test, forecast_test_auto)
mape = mean_absolute_percentage_error(df_test, forecast_test_auto)
rmse = np.sqrt(mean_squared_error(df_test, forecast_test_auto))
print(f'mae - auto: {mae}')
print(f'mape - auto: {mape}')
print(f'rmse - auto: {rmse}')
```

ДОДАТОК В (презентація)

Прогнозування фінансових часових рядів. Порівняльний аналіз методів прогнозування

Підготував: Макітрук Максим КА-02

06.06.2024

Вступ

Метою дослідження є побудова прогнозування та порівняння результатів дослідження для різних моделей прогнозування.

Важливість прогнозування полягає у тому що це відкриває можливості для управління ризиками, оптимізування портфеля інвестора, для прийняття обґрунтованих рішень на основі математичних даних, покращення торговельних стратегій.

Тобто прогнозування корисне у всіх видах прийняття рішень, що стосуються майбутнього руху даних.

Методи дослідження

У своїй роботі я використовую дві найбільш популярні моделі прогнозування фінансових часових рядів, а саме модель ARIMA та LSTM, обидві моделі можуть враховувати сезонність даних та їх тренди, що дозволяє оптимально та з достатньою точністю прогнозувати майбутні дані на основі бази попередніх історичних значень.

Саме дві моделі розглядаються через різні принципи роботи, що включають суто математичну модель ARIMA та модель, що базується на машинному навчанні, LSTM.

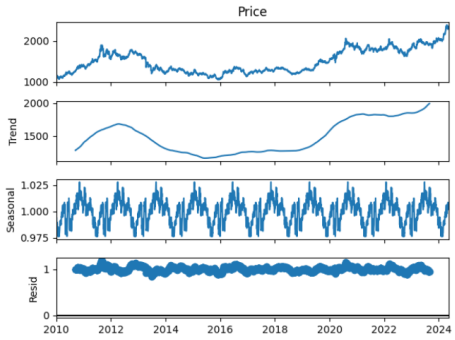
Підготовка даних

Перед тим як програмний апарат зможе опрацювати інформацію та зробити прогноз потрібно впевнитись що інформація є повною та якісною, для цього рекомендую користуватись спеціалізованими сайтами, що надають інформацію з бірж та сайтом Kagle, що надає обширні датасети для аналізу.

У самій програмі я реалізував відсіювання непотрібної інформації та підготовку даних до вигляду з яким комфортно працювати, наділяючи усі важливі дані правильними форматами, аби методами опрацювання інформації наявними у бібліотеках мови Python можна було оптимально прауювати.

Сезонна декомпозиція

Для кращого розуміння інформації, її трендів та сезонності варто застосувати сезонну декомпозицію даних, у моїй роботі вона має вигляд :



З графіку видно що дані мають досить сильно виражений тренд, але сезонність не чітка та існує чимало залишків, які не відповідають сезонності.

Модель ARIMA

Модель авторегресії AR(p) $y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_p y_{t-p} + \varepsilon_t$

Модель авторегресії з ковзним середнім ARMA(p,q)

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_p y_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + b_2 \varepsilon_{t-2} + \cdots + b_q \varepsilon_{t-q}$$

Інтегрована модель авторегресії з ковзним середнім ARIMA(p,d,q)

$$\Delta^d y_t = a_0 + a_1 \Delta y_{t-1} + a_2 \Delta y_{t-2} + \cdots + a_p \Delta y_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + b_2 \varepsilon_{t-2} + \cdots + b_q \varepsilon_{t-q},$$

Модель множинної лінійної регресії $y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_p x_p + \varepsilon$

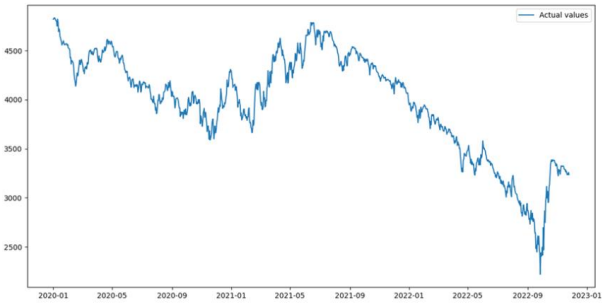
Модель ARIMA

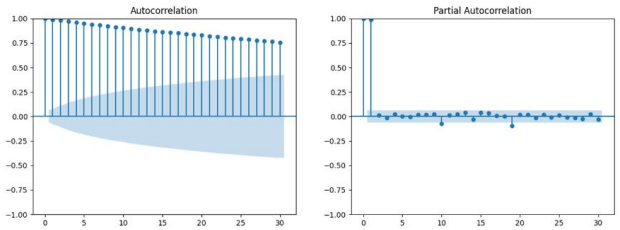
Модель ARIMA містить у собі три параметри (p,d,q), а саме p-авторегресія, d-інтегрування, q-ковзке середнє, на початку дослідження завжди обирається параметр d, адже його вибір базується на кількості диференціювань даних моделі для досягнення стаціонарності часового ряду.

Для підбору параметрів я розглядаю дві альтернативи, а саме:

1) аналіз графіків ACF та PACF (автокореляційні функції) та використання тесту Дікі-Фуллера
2) Розумний перебір з мінімалізацією критеріїв якості моделі AIC та SIC (критерій Айке та критерій Шварца)

Після самого підбору параметрів потрібно розбити на тренувальну частину та тестову для перевірки результатів, я розбиваю на частини 95% та 5%. Після побудови підрахуємо відхилення та відсоток надійності.






```
P_Value :  0.46630799407758927
```

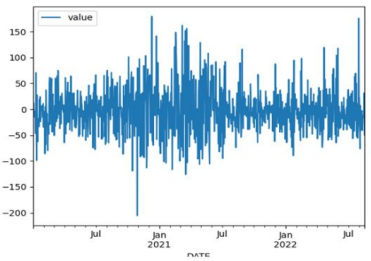
Після диференціювання

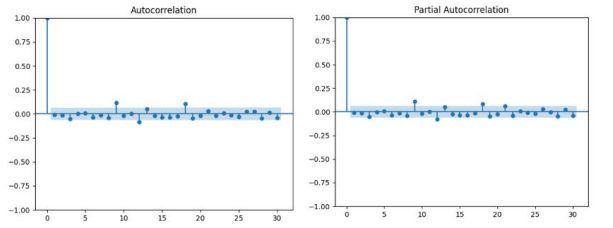
після диференціювання бачимо що показник стає меншим за 0.05, що є дуже важливим для признання ряду стаціонарним, далі пармерти p та q обираються наступним чином :

Якщо графік PACF має значущий сплеск на лагах p, але не далі, а графік ACF спадає поступово, це може свідчити про модель ARIMA(p, d, 0).

Якщо графік ACF має значущий сплеск на лагах q, але не далі, а графік PACF спадає поступово, це може свідчити про модель ARIMA(0, d, q).

У моєму прикладі таких явних показників не виявлено, тому виконувався розумний перебір.

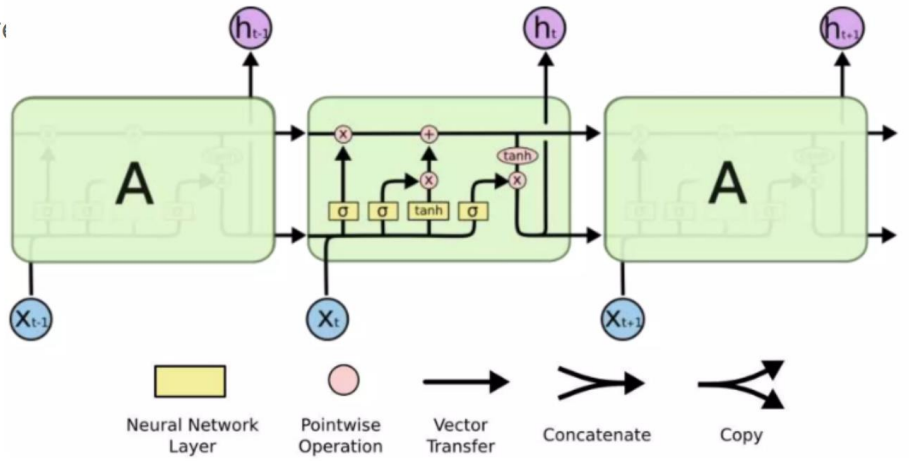





```
P_Value :  2.7531274485120787e-09
```

Графічне представлення LSTM

Введите т



Модель LSTM

Принцип роботи LSTM

LSTM (Long Short-Term Memory) – це спеціалізована рекурентна нейронна мережа (RNN), призначена для обробки послідовностей даних та запам'ятовування довготривалих залежностей. Головна особливість LSTM – комірки пам'яті з вхідним, вихідним та забуваючим гейтами, що дозволяють ефективно зберігати і керувати інформацією. Вхідний гейт контролює, яку нову інформацію додати, забуваючий – яку видалити, а вихідний – яку використовувати для генерації виходу.

Архітектура моделі

Модель LSTM побудована з двома LSTM шарами та одним Dense шаром. Перший LSTM шар обробляє вхідні дані та передає їх до другого LSTM шару для врахування довготривалих залежностей. Після обробки даних LSTM шарами, Dense шар використовується для перетворення результатів у прогнозовані значення.

Процес навчання моделі

Навчання моделі LSTM включає масштабування даних до діапазону [0, 1], розділення на навчальний та тестовий набори, та навчання за допомогою зворотного поширення помилки. Використовувався оптимізатор Adam та функція втрат mean_squared_error. Після навчання модель прогнозує на тестових даних, а результати оцінюються за допомогою метрик MSE, MAE та MAPE.

Метрики оцінки якості

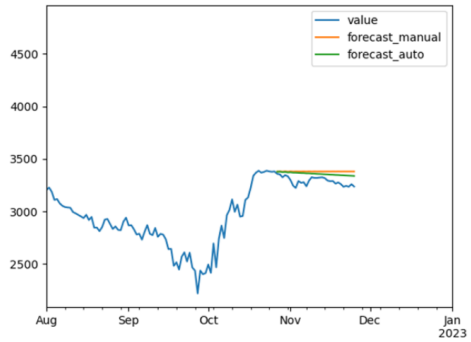
1.MAE (Mean Absolute Error) – вимірює середню абсолютну похибку між прогнозованими та фактичними значеннями. Це дає уявлення про середню похибку прогнозів.

2.MAPE (Mean Absolute Percentage Error) – вимірює середню абсолютну відносну похибку між прогнозованими та фактичними значеннями. Це корисно для оцінки точності моделей у відсотковому вираженні.

3.RMSE (Root Mean Squared Error) – вимірює корінь середнього квадратичного відхилення між прогнозованими та фактичними значеннями. Це показник, що акцентує увагу на великих похибках, роблячи його корисним для виявлення великих відхилень у прогнозах.

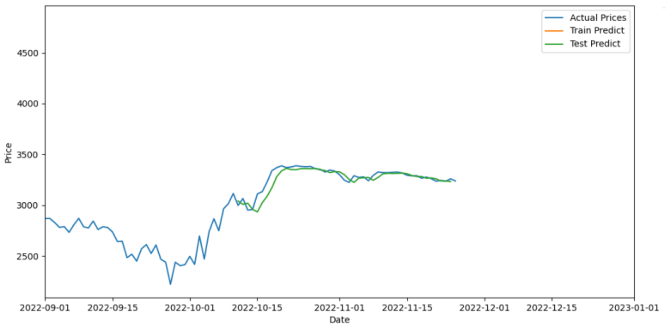
Результати прогнозування руху індексу S&P 500

ARIMA




```
mae - manual: 90.967279196524
mape - manual: 0.02779643211679418
rmse - manual: 98.67158877203413
mae - auto: 69.30349180374836
mape - auto: 0.021184751496169843
rmse - auto: 76.70271531493785
```

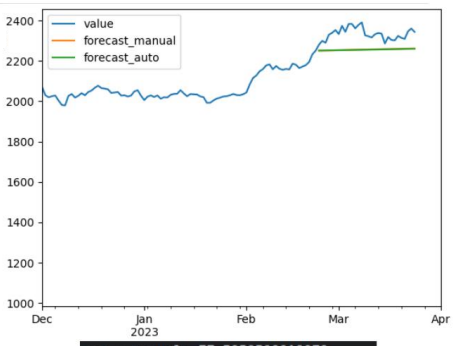
LSTM




```
mae : 35.16004752604168
mape : 0.01086407051149407
rmse : 54.52000781099759
```

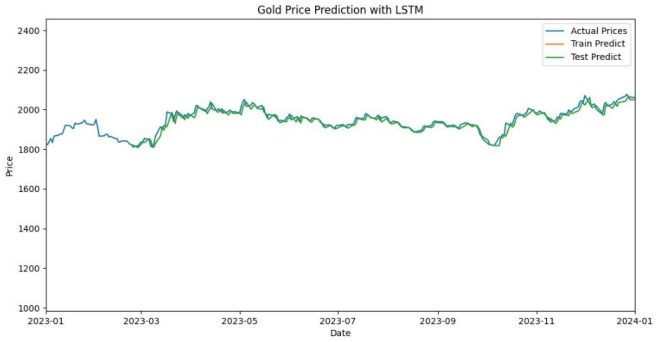
Результати прогнозування цін на золото

ARIMA




```
mae - manual: 77.3850508819979
mape - manual: 0.033008413316499556
rmse - manual: 82.90039457857189
mae - auto: 78.32818781497608
mape - auto: 0.03341276529206803
rmse - auto: 83.75434714070562
```

LSTM




```
mae : 17.62069686813598
mape : 0.00847839903157342
rmse : 24.380956930363283
```

Порівняння результатів для індексу S&P 500

Результати	Arima вручну	Arima автоматична	LSTM
MAPE	2.7%	2.1%	1%
MAE	90.96	69.30	35.16
RMSE	98.6	76.7	54.52
Проміжок прогнозування	1 місяць	1 місяць	1 місяць

Порівняння результатів для цін на золото

Результати	Arima вручну	Arima автоматична	LSTM
MAPE	3,3%	3,3%	0,8%
MAE	77,38	78,32	17,62
RMSE	82,9	83,75	24,38
Проміжок прогнозування	1 місяць	1 місяць	1 місяць

Дякую за увагу !