НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

«На правах рукопису»

УДК 004.852

До захисту допущено:

Завідувач кафедри

_____ О.Л. ТИМОЩУК

«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Системний аналіз фінансового ринку»

спеціальності 124 «Системний аналіз»

на тему: «Застосування альтернативних методів для моделювання і прогнозування фінансових процесів у банківській сфері»

Виконала:
студентка ІІ курсу, групи КА-02мп
Садома Юлія Володимирівна _____

Керівник:
професор кафедри ММСА,
д.т.н., проф., Бідюк Петро Іванович _____

Рецензент::
декан ФІОТ КПІ ім. Ігоря Сікорського
д.т.н., проф.,Теленик С.Ф. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ
2021

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – другий (магістерський)
Спеціальність – 124 «Системний аналіз»
Освітньо-професійна програма «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Оксана ТИМОЩУК
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Садомі Юлії Володимирівні

1. Тема роботи «Застосування альтернативних методів для моделювання і прогнозування фінансових процесів у банківській сфері», керівник роботи Бідюк Петро Іванович, д.т.н., професор, затверджені наказом по університету від «_02_»_листопада_ 2021 р .№_3651-с__

2. Термін подання студентом роботи 12 грудня 2021 року _____

3. Об’єкт дослідження: часові ряди фінансово-економічних показників.

4. Предмет дослідження: математичні моделі та методи для прогнозування та аналізу даних

5. Перелік завдань, які потрібно розробити

1. Проаналізувати існуючі методи, що використовуються для прогнозування даних, фільтрації даних та заповнення пропусків
2. Обрати методи для прогнозування даних, фільтрації даних та заповнення пропусків

3. Розробити програмний продукт на базі обраних математичних методів
4. Застосувати розроблений програмний продукт на реальних даних, проаналізувати результати
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):
6. Орієнтовний перелік публікацій:

   Садома Ю. В. Застосування альтернативних методів для моделювання і прогнозування фінансових процесів у банківській сфері, VIII Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 25-26 листопада 2021 року Київ.

7. Дата видачі завдання: 5 вересня 2021

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання магістерської дисертації	Примітка
1	Отримання завдання	01.09.2021 – 05.09.2021	Виконано
2	Збір інформації	06.09.2021 – 12.09.2021	Виконано
3	Ознайомлення з літературою і підготовка теоретичної частини магістерської дисертації	13.09.2021 – 26.09.2021	Виконано
4	Аналіз вимог завдання, вибір методів і засобів розв’язання поставленої задачі	27.09.2021 – 3.10.2021	Виконано
5	Розробка програмного продукту	4.10.2021 – 31.10.2021	Виконано
6	Проведення аналізу ринкових можливостей стартап-проекту	1.11.2021 – 7.11.2021	Виконано

7	Отримання висновків після тестування	8.11.2021 – 15.11.2021	Виконано
8	Оформлення магістерської дисертації	15.11.2021 – 12.12.2021	Виконано
9	Отримання допуску до захисту та подача роботи до ДЕК	13.12.2021 – 14.12.2021	Виконано

Студент Юлія САДОМА

Керівник Петро БІДЮК

РЕФЕРАТ

Магістерська дисертація: 106 с., 59 рис., 28 табл., 1 дод., 9 джерел.

ФІНАНСОВО-ЕКОНОМІЧНІ ПОКАЗНИКИ, КОЕФІЦІЄНТ ШВИДКОЇ ЛІКВІДНОСТІ, РЕНТАБЕЛЬНІСТЬ АКТИВІВ, АРКС, АРІКС, СЕЗОННА АРІКС, МНОЖИННА РЕГРЕСІЯ, ГРАДІЄНТНИЙ БУСТИНГ, НЕЙРОННІ МЕРЕЖІ

Об’єкт дослідження: часові ряди фінансово-економічних показників.

Предмет дослідження: математичні моделі та методи для прогнозування та аналізу даних.

Метою роботи є створення та реалізація програмного продукту для аналізу та прогнозування фінансово-економічних показників, що впливають на стабільність роботи банку.

Результатом роботи є програмний продукт написаний мовою програмування Python. Створений програмний продукт може спростити деякі аспекти роботи працівників фінансових установ та бути корисним у банківській сфері.

Подальший розвиток предмету дослідження – розгляд більш складних методів та для прогнозування та аналізу даних.

ABSTRACT

Master’s thesis explanatory note: 109 p., 59 fig., 28 tables, 1 appendixes, 9 sources.

FINANCIAL AND ECONOMIC INDICATORS, QUICK LIQUIDITY RATIO, RETURN ON ASSETS, ARMA, ARIMA, SEASONAL ARIMA, PLURAL REGRESSION, GRADIENT BUSTING, NEURAL NETWORKS

The object of research: time series of financial and economic indicators.

The subject of research: mathematical models and methods for forecasting and data analysis.

The purpose of the work is to create and implement a software product for analysis and forecasting of financial and economic indicators that affect the stability of the bank.

The result is a software product written in the Python programming language. The created software product can simplify some aspects of the work of employees of financial institutions and be useful in the banking sector.

Further development of the subject of research - consideration of more complex methods for forecasting and data analysis.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СП – семантичне правило.

ФН – функція належності.

НМ – нейронні мережі.

ПЗ – програмне забезпечення.

ПП – програмний продукт.

ЛР – лінійна регресія.

ОС – операційна система.

АЧР – аналіз часових рядів.

SAS – Statistical Analysis System.

Eviews – Econometric Views.

VaR – Value-at-Risk.

ВВ – випадкова величина.

MSE - Mean square error.

MAE – Mean absolute error.

MAPE – Mean absolute percentage error.

ВСТУП

На сьогоднішній день найважливішими фінансовими інститутами для ринкової економіки виступають банки. У порівнянні з будь-якими іншими фінансовими установами, вони надають найбільше споживчих кредитів, при чому така тенденція зберігається і на світовому ринку і на національному. Банк у свою чергу, є фінансовим інститутом, який пропонує найширший спектр послуг, що перш за все відносяться до кредитів, заощаджень і платежів, і виконують різноманітні фінансові операції.

Крім того, банки вважаються найважливішою частиною ринкової економіки. У їх діяльності задіяна величезна частина грошового обороту в державі, формуються джерела капіталу у результаті чого виконується перерозподіл тимчасово звільнених грошових коштів. Крім того, варто зазначити що комерційні банки беруть участь у процесі переходження грошового капіталу від менш ефективних галузей у національній економіці до найбільш конкурентоспроможних. Комерційні банки також беруть участь у процесі акумуляції тимчасово вільних грошових засобів різноманітних компаній, підприємств, організацій, установ та безпосередньо держави і передачі грошових коштів зі сфер накопичення у так звані сфери використання.

Комерційні банки як елемент банківської системи сприяють економії суспільних витрат звернення, сприяючи прискоренню обороту грошей прискореним розрахунком, переказу грошей випуском кредитних знарядь замість готівки, наприклад, векселів, чеків, дебетових і кредитних карток, сертифікатів і т.д. Діяльність банку як вираз його економічних відносин з клієнтами визначається його сутністю, функціями і роллю в економіці.

Проте, разом з тим, існує безліч факторів, що можуть викликати банкрутство банка. Що, у свою чергу, може запустити цілу низку банкрутств підприємств, банків і приватних осіб що мають відношення до банку. Тому,

останнім часом приділяється все більше уваги детальному аналізу та зваженій оцінці основних фінансових показників з боку банків та інших кредитних установ.

Метою даної магістерської дисертації є створення та реалізація програмного продукту для аналізу та прогнозування фінансових показників, що у свою чергу можуть гарантувати стабільність банку та попередити або ж навіть запобігти несприятливій ситуації. Створений ПЗ може спростити деякі аспекти роботи банку та бути корисним у банківській сфері.

З огляду на вищесказане, можемо зробити висновок, що фінансові показники достовірно відображають стан банку, а їх вчасна оцінка, аналіз, прогнозування та управління ними – важливий аспект стратегії розвитку та стабільної роботи будь-яких кредитних установ, саме це зумовлює актуальність обраної теми магістерської дисертації.

РОЗДІЛ 1 ФІНАНСОВІ ПОКАЗНИКИ ТА ПІДХОДИ ДО ЇХ ОБЧИСЛЕННЯ ТА АНАЛІЗУ

1.1 Поняття та класифікація фінансових показників

1.1.1 Рентабельність активів

Рентабельність активів (return on assets, ROA) — це тип показника рентабельності інвестицій (ROI), який вимірює прибутковість бізнесу по співвідношенню з його сукупними активами. Тобто даний коефіцієнт показує, наскільки добре працює компанія, порівнюючи прибуток (чистий прибуток), який вона генерує, з капіталом, який вона інвестує в активи. Чим вища віддача, тим продуктивніше й ефективніше управління у використанні економічних ресурсів. Основна формула розрахунку показника Return on Assets (ROA) базується на співвідношенні чистого прибутку і сумарних активів:

$$ROA = \frac{Net\ Profit}{Total\ Assets} \times 100\%, \quad (1.1)$$

де Net Profit - річний чистий прибуток,

Total Assets - середньорічна величина сумарних активів компанії (іноді в розрахунках також використовується просто сума активів на кінець року). Коефіцієнт ROA показує ефективність використання капіталу, задіяного в діяльності компанії. Сумарні активи в балансі завжди рівні сумарних зобов'язань, тому значення в знаменнику ROA можна інтерпретувати і як активи, і як всі зобов'язання і капітал, залучені для ведення бізнесу.

Величину ROA можна порівнювати із середньозваженою вартістю капіталу компанії або з необхідною прибутковістю її акціонерного капіталу, але в обох випадках треба враховувати, що рентабельність власного капіталу не

зовсім точно відображає ці процентні показники. У порівнянні з прибутковістю власного капіталу відмінність полягає в тому, що знаменник ROA включає всі активи, в тому числі і ті, які були профінансовані позиковим капіталом. Отже, для ROA цілком припустимі значення менше, ніж необхідна прибутковість на власний капітал.

1.1.2 Норматив достатності власних коштів

Норматив достатності власних коштів (капіталу) Н1 – напевно найголовніший норматив, слідувати якому повинні усі фінансові установи а також кредитні організації. Даний норматив найкраще може продемонструвати надійність банку, адже демонструє здатність банку обходити теоретично можливі фінансові втрати за свій рахунок, без заподіяння фінансової шкоди своїм клієнтам.

Крім того Н1 демонструє, незалежність організації від кредиторів. Якщо значення коефіцієнта мале, то така організація сильно залежить від позикових джерел фінансування, та має доволі нестійке фінансове становище. Мінімальне значення нормативу становить 10,0%.

1.1.3 Коефіцієнти ліквідності

Коефіцієнти ліквідності - це показник, який вказує на здатність людини погасити свій борг, як і коли вони настають. Іншими словами, можна сказати, що цей коефіцієнт показує, як швидко компанія може перетворити свої поточні активи в готівку, щоб вона могла своєчасно погасити свої зобов'язання. Як правило, ліквідність і короткострокова платоспроможність використовуються разом. Крім того коефіцієнт ліквідності впливає безпосередньо на довіру до компанії, а також на її кредитний рейтинг. Якщо є постійні непогашення короткострокового зобов'язання, це призведе до банкрутства. Тому цей коефіцієнт відіграє важливу роль у фінансовій стабільності будь-якої компанії та кредитних рейтингах.

Коефіцієнт поточної ліквідності вимірює фінансову міцність компанії. Як правило, 2:1 розглядається як ідеальне співвідношення, але воно залежить у першу чергу від галузі.

$$К_{пл} = \frac{ПА}{КЗ}, \tag{1.2}$$

де ПА – сума поточних активів,

КЗ – сума короткострокових зобов'язань.

Коефіцієнт швидкої ліквідності є найкращим показником для виміру ліквідності компанії. Він є більш консервативним, ніж коефіцієнт поточної ліквідності. Коефіцієнт швидкої ліквідності обчислюється шляхом коригування оборотних активів, щоб виключити ті активи, яких немає в готівці. Як правило, 1:1 вважається ідеальним співвідношенням для даного показника.

Коефіцієнт абсолютної ліквідності використовується для виміру загальної ліквідності, доступної для компанії. Він враховує лише ринкові цінні папери та грошові кошти, доступні компанії. Цей коефіцієнт перевіряє лише короткострокову ліквідність з точки зору готівки, ринкових цінних паперів та поточних інвестицій.

Основний коефіцієнт захисного інтервалу вимірює кількість. днів, коли підприємство може покрити свої грошові витрати без допомоги додаткового

1.1.4 Чистий оборотний капітал

Чистий оборотний капітал (Net Working Capital) – це різниця між поточними активами підприємства та його поточними зобов'язаннями на її балансі. Це показник ліквідності компанії та її здатності виконувати короткострокові зобов'язання, а також фінансувати операції бізнесу. Ідеальна позиція – мати більше поточних активів, ніж поточних зобов'язань, і, таким чином, мати позитивний баланс чистого оборотного капіталу. Формула розрахунку чистого розрахункового капіталу має вигляд:

$$\text{NWC} = \text{ПА} - \text{ПЗ}, \tag{1.3}$$

де ПА – сума поточних активів,

КЗ – сума поточних зобов'язань.

Збільшення величини NWC може свідчити про підвищення показника ліквідності підприємства і зростання його кредитоспроможності. Хоча, занадто

великі значення NWC – свого роду сигнал про не надто ефективну фінансову політику підприємства, що у свою чергу може призвести до зниження рентабельності.

1.1.5 Коефіцієнт покриття активів

Коефіцієнт покриття активів демонструє, наскільки добре організація може сплатити свої борги. Його використовують сторонні аналітики, такі як кредитори та інвестори, під час проведення експертизи фінансів бізнесу. Зокрема, кредитор хоче, щоб цей коефіцієнт перевищив мінімальний пороговий рівень, перш ніж він погодиться надати гроші позичальнику. Він може захотіти побачити історію, коли позичальник міг постійно перевищувати цей мінімальний пороговий рівень.

ACR за своєю природою схожий на коефіцієнт покриття боргу, але розглядає активи балансу замість порівняння доходу з рівнем боргу. Коефіцієнт обчислюється за даною формулою:

$$ACR = Total\ Assets - \frac{Short - term\ Liabilities}{Total\ Debt}, \quad (1.4)$$

Cash flow to debt ratio – відношення грошового потоку до боргу, як випливає з назви, порівнює загальний грошовий потік із загальною заборгованістю компанії. Це один із коефіцієнтів покриття, які використовуються у фінансовому світі для перевірки стану компанії. Коефіцієнт говорить про платоспроможність компанії.

$$Cash\ Flow\ to\ Debt = \frac{Cash\ Flow\ from\ Operations}{Total\ Debt}, \quad (1.5)$$

1.2 Підходи математичного моделювання, що використовуються для прогнозування показників

1.2.1 Регресійний підхід

Регресійний аналіз – це причинно-наслідковий/економетричний метод прогнозування. Деякі методи прогнозування засновані на припущенні, що можна визначити основні фактори, що мають вплив на прогнозовану змінну.

Тобто регресійні моделі поділяють змінні на пояснюючі та результуючі і розглядають їх взаємозв'язок. Регресійний аналіз включає велику групу методів, які можна використовувати для прогнозування майбутніх значень змінної, використовуючи інформацію про інші змінні. Ці методи включають як параметричні (лінійні чи нелінійні), так і непараметричні методи. Класичні припущення для регресійного аналізу включають:

1. Вибірка є репрезентативною для сукупності для передбачення висновку.
2. Помилка є випадковою величиною з нульовим середнім, що залежить від пояснювальних змінних.
3. Незалежні змінні вимірюються без помилок.
4. Помилки є некорельованими, тобто матриця дисперсія – коваріація помилок є діагональною, а кожен ненульовий елемент є дисперсією помилки.
5. Дисперсія похибки є постійною в межах спостережень.

Суттєвими перевагами регресійних над іншими типами моделей виступає їх швидкість, гнучкість та простота. До недоліків можна віднести складність визначення параметрів.

1.2.2 Логістична регресія

Логістична регресія – це різновид множинної регресії, який слід використовувати у випадку, коли залежна змінна є дихотомічною (бінарною). Логістична регресія є прогнозним аналізом. Вона використовується для опису даних і пояснення зв'язку між однією залежною двійковою змінною та однією або кількома номінальними, порядковими, інтервальними або незалежними змінними на рівні співвідношення.

Замість підгону прямої лінії або гіперплощини модель логістичної регресії використовує логістичну функцію для стиснення результату лінійного рівняння між 0 та 1. Логістична функція визначається як:

$$P = \frac{1}{1 + e^{-y}}, \tag{1.6}$$

де P – ймовірність виникнення цікавої для нас події;
e – основа натурального логарифму;
y – деяка змінна, що визначається рівнянням ЛР.

1.2.3 Нечіткі нейронні мережі

Головною особливістю нечітких нейронних мереж порівняно з іншими підходами є представлення вектору даних у формі лінгвістичних змінних, з метою дослідження крім кількісних ще й якісних характеристик. Дані змінні можуть бути представлені наступним чином:

$$<b,T,X,G,M>, \tag{1.7}$$

де b – ім’я даної лінгвістичної змінної;

T –множина змінної, яка у свою чергу складається з термів. Терм представляє собою нечітку множину на універсальній множині X;

G - синтаксичне правило;

M – семантичне правило, яке задає функцію належності.

Семантичне правило будується лише після того як побудовані функції належності для усіх лінгвістичних змінних.

Підхід, на основі нечітких нейромереж, автоматизує побудову функцій належності, що у свою чергу автоматизує побудову база знань і логічного висновку.

Далі показано загальний вигляд алгоритму НМ:

1. Визначення множини вхідних змінних: X.
2. Визначення множини вихідних змінних: Y.
3. Формування множини T, що складається з функцій належності кожної змінної.
4. Формування кінцевої множини нечітких правил.
5. Визначення істинності для кожного правила.
6. Визначення нечітких підмножин для вихідних змінних.
7. Формування композиції нечітких підмножин вихідних змінних.
8. Формування чіткого значення безпосередньо для кожної з ЛЗ виходу.

1.3 Огляд існуючих комп’ютерних систем для побудови моделей фінансово-економічних процесів

1.3.1Eviews

EViews (Econometric Views) — являє собою пакет економетрики, статистики та прогнозування, який допомагає реалізовувати потужні аналітичні інструменти в інтуїтивно зрозумілому, простому у використанні інтерфейсі.

За допомогою EViews, з’являється можливість швидко й ефективно виконувати різного роду операції зі своїми даними, а саме виконувати економетричний або статистичний аналіз, будувати прогнози чи моделі, а також виводити чіткі графіки та таблиці для демонстрації результату. Пакет економетрики Econometric Views розроблений компанією Quantitative Micro Software (QMS), перша версія якого була випущена у березні 1994-го року. На даний момент найновішою стабільною версією Econometric Views є 12-та, яка була випущена у листопаді 2020-го року. Особливості інтерфейсу Econometric Views можна спростерігати на рисунках 1.1-1.2.

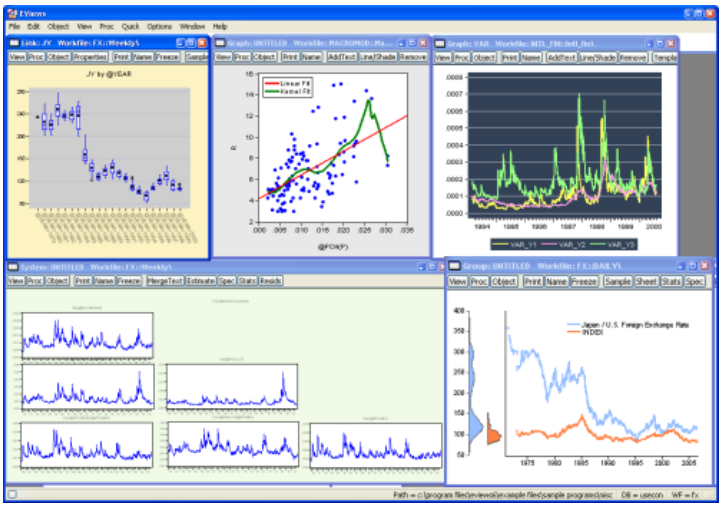


Рисунок 1.1 – Інтерфейс 1 Econometric Views

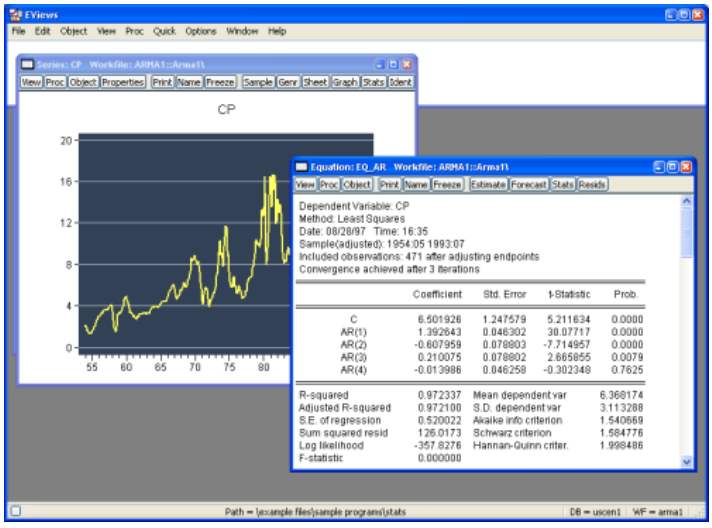


Рисунок 1.2 – Інтерфейс 2 Econometric Views

Переваги:

- відносно простий у розумінні а також використанні;
- має інтуїтивно зрозумілий, гнучкий інтерфейс.

Недоліки:

- Econometric Views орієнтований виключно на операційну систему Windows;
- Econometric Viewsатентований є патентовим програмним забезпеченням, ціна на який варіюється від $1,700 до $2,000.

1.3.2 R

R — це мова програмування та безкоштовне програмне забезпечення, розроблене Россом Іхакою та Робертом Джентльменом у 1993 році.

У мові програмування R присутній великий каталог статистичних та графічних методів, а саме: алгоритми машинного навчання, лінійна регресія, часові ряди, статистичний висновок. Більша частина бібліотек R написані на R, але для надскладних обчислювальних завдань перевага надається кодам написаним на мовахC, C++ і Fortran.

Аналіз даних за допомогою R відбувається в декілька кроків: програмування, перетворення, виявлення, моделювання та передача результатів. На даний момент найновішою версією R є 4.1.2, яка була випущена 1 листопада 2021-го року. Інтерфейс R можемо спостерігати на рисунку 1.3.

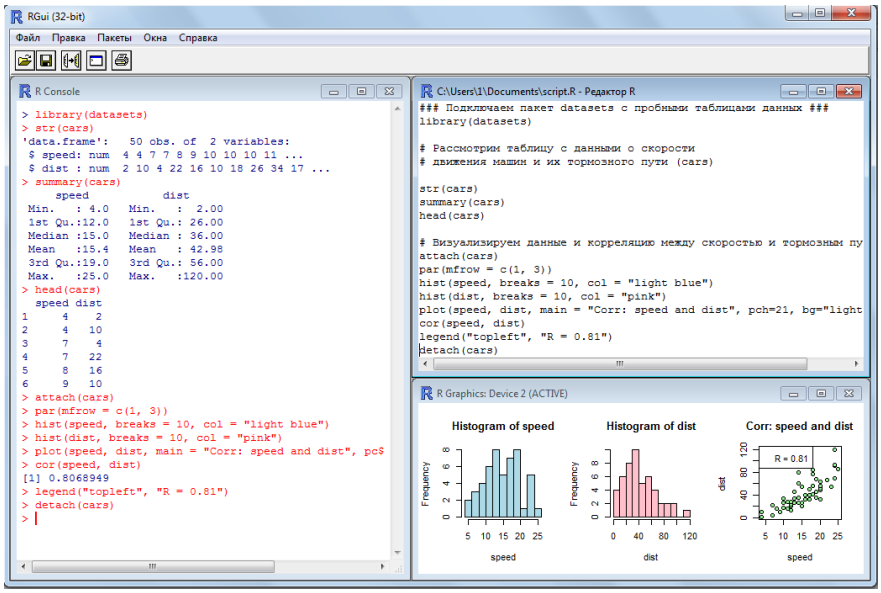


Рисунок 1.3 – Інтерфейс R Studio

Переваги мови програмування R:

- сумісний з будь-якою операційною системою;
- синтаксис зрозумілий і легкий з точки зору вивчення;
- існує безліч додаткових пакетів (більш ніж 7000) для розширення функціоналу;
- R є повноцінною та цілісною мова програмування;
- повністю безкоштовне, знаходиться вільному доступі.

Недоліки мови програмування R:

- у користувачів незнайомих з мовами програмування та програмним кодом, можуть виникнути певні труднощі у використанні.

1.3.3 SAS

SAS (Statistical Analysis System)- це статистичне програмне середовище основною функцією якого є робота з великими масивами даних, а саме їх аналіз, обробка, управління і прогнозування. Розробкою програмного забезпечення

займався Інститут SAS ще у 1976 році. На даний момент найновішою стабільною версією SAS є 9.4M7, яка була випущена 18 серпня 2020-го року. Інтерфейс ПЗ SAS можемо спостерігати на рисунку 1.4.

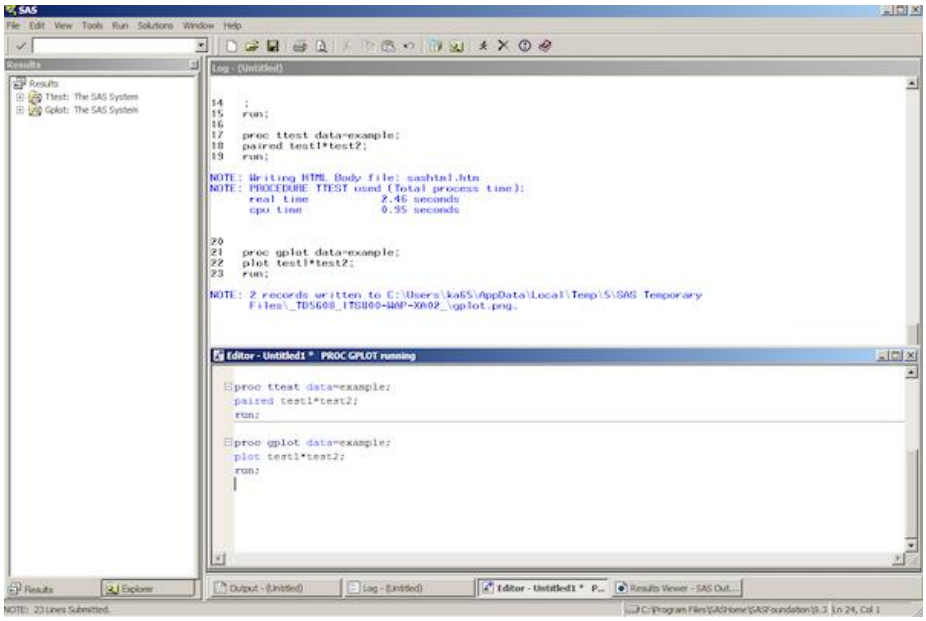


Рисунок 1.4 – Інтерфейс SAS

Переваги програмного забезпечення SAS:

- сумісний з будь-якою операційною системою;
- простий та зрозумілий синтаксис написання коду;
- вихідні дані подаються у зручному для роботи форматі;
- сумісний з базами даних різного типу;
- доволі просте налагодження та редагування коду;

Недоліки програмного забезпечення SAS:

- ПЗ не є безкоштовним, необхідна покупка ліцензії на користування (приблизно $9000.00 на рік);
- з точки зору об’єму програмного коду порівняно з мовою програмування R SAS менш продуктивний;
- у SAS документація більшості алгоритмів не є публічною, що у свою чергу дещо обмежує роботу та вивчення ПЗ.

1.3.4 MATLAB

MATLAB(matrix laboratory) – обчислювальне середовище, яке дозволяє працювати з матрицями, будувати графіки функцій і даних, реалізовувати алгоритми, створювати інтерфейси користувача та взаємодіяти з програмами, написаними іншими мовами. Пакет прикладних програм MATLAB було створено групою розробників компанії The MathWorks. На даний момент найновішою версією MATLAB є R2021b, яка вийшла 22 вересня 2021-го року. Інтерфейс MATLAB можемо спостерігати на рисунку 1.5.

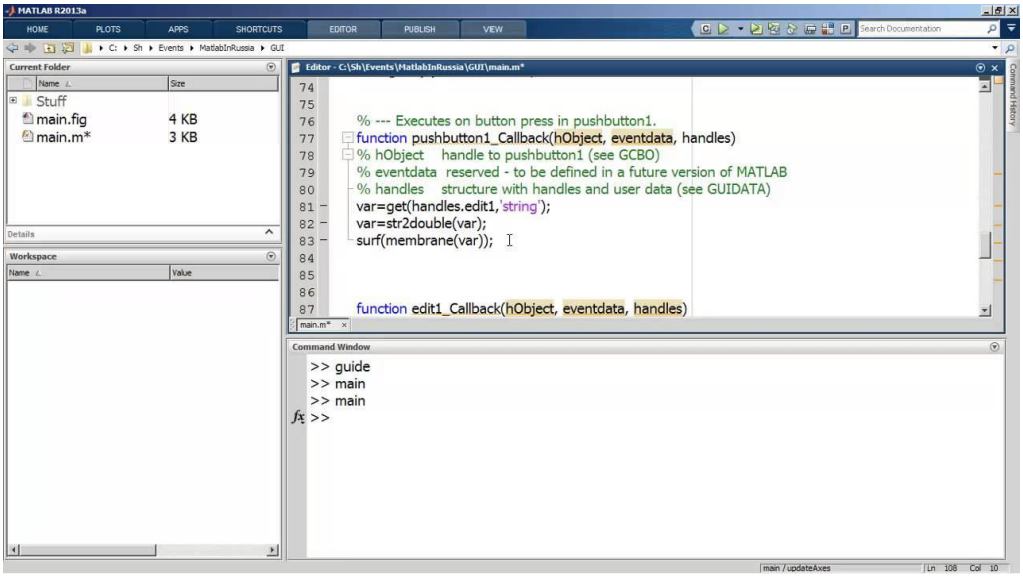


Рисунок 1.5 – Інтерфейс MATLAB

Переваги MATLAB:

- доволі легкий синтаксис з точки зору вивчення;
- наявна конвертація у код мовою C або C ++;
- обширний різноманітний функціонал.

Недоліки MATLAB:

- для роботи потребує купівлю ліцензії ($1,500-$3000);
- вузьконаправленість мови програмування;

1.3.5 Порівняння комп’ютерних систем для побудови моделей фінансово-економічних процесів

Проаналізувавши всю зібрану інформацію та провівши порівняльний аналіз вищерозглянутих комп’ютерних систем, внесемо основні моменти у таблицю 1.1.

Таблиця 1.1 – Результати порівняльного аналізу ПЗ

Назва ПЗ	Багатофункціональність ПЗ	Складність написання коду	ОС	Вартість
Eviews	-	-	Microsoft Windows	$1,700-$2,000
R	+	+-	GNU/Linux, Microsoft Windows, macOS BSD	—
SAS	+	-	Microsoft Windows IBM mainframe Unix/Linux OpenVMS Alpha	$9000.00 на рік
MATLAB	-	-	Microsoft Windows Unix/Linux	$1,500-$3000

1.4 Висновки до розділу та постановка задачі дослідження

У даному розділі було розглянуто такі показники, як «рентабельність активів», «норматив достатності власних коштів», «коефіцієнти ліквідності» а також «чистий оборотний капітал», способи їх обчислення та їх трактування відносно ситуації у банку. Також було наведено декілька популярних підходів до прогнозування даних показників. Крім того на основі знайденої інформації була заповнена порівняльна таблиця, що охоплює існуючі комп'ютерні системи для побудови моделей фінансово-економічних процесів. Проаналізувавши яку, можна помітити, що на сьогоднішній день найзручнішим ПЗ а також мовою програмування – є R.

РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ТА АНАЛІЗУ ДАНИХ

2.1 Основні поняття

Для побудови економетричних моделей зазвичай використовують наступні типи даних:

- дані, які відображають певну низку процесів у певний момент часу t;
- дані, які відображають один певний процес вподовж деяких послідовних періодів часу t.

Просторові моделі будуються по даних першого типу, а по даних другого – моделі часових рядів.

Часовий ряд $y_1,\ldots,y_k,\ldots,y_t$ це відображення значення якогось процесу, об'єкту чи ознаки, які вимірюють впродовж певних постійних інтервалів часу t.

Основними компонентами часового ряду є:

- тренд;
- сезонність;
- цикл;
- похибка.

Розрізняють стаціонарні (рисунок 2.1) та нестаціонарні (рисунок 2.2) ЧР. Ряд $y_1, \ldots, y_T$, вважається стаціонарним якщо виконується умова: для будь-якого $\forall s \;\; y_t, \ldots, y_{t+s}$ не залежить від часу t. Тобто, властивості даного ряду жодним чином не мають залежності від t.

У нестаціонарних часових рядів відсутня дана властивість. Варто відмітити, що нестаціонарний ЧР легко помітити за наявністю тренду чи сезонності. А от наявність циклів ще не говорить про нестаціонарність досліджуваного ряду.

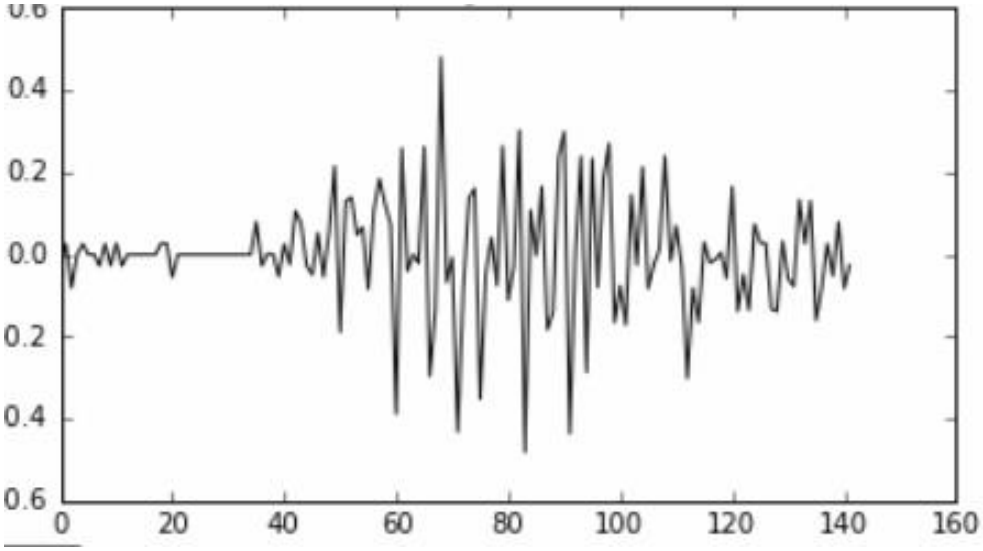


Рисунок 2.1 – Приклад стаціонарного часового ряду

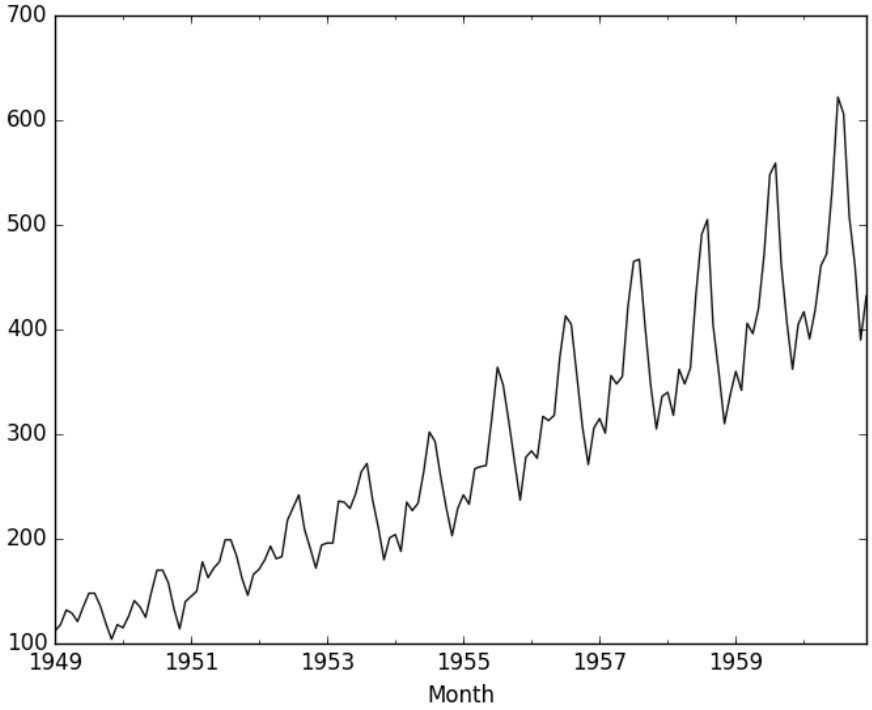


Рисунок 2.2 – Приклад нестаціонарного часового ряду

Загалом існує величезна кількість тестів що перевіряють стаціонарність ЧР. Тест Діккі-Фуллера (Augmented Dickey-Fuller test) – найпоширеніший тест для перевірки наявності одиничного кореня. Під час тесту відбувається перевірка гіпотез. Нульовою гіпотезою H0 у тесті Діккі-Фуллера виступає наявність

одиничного кореня, що у свою чергу свідчить про те, що досліджуваний ряд нестаціонарний. Н1 – гіпотеза щодо стаціонарності часового ряду. Статистикою тесту Діккі-Фуллера є класична t-статистика, у даному випадку її розподілом виступає вінерівський процес, названий розподілом Діккі-Фуллера.

Розрізняють 3 види тестової регресії:

$$\Delta y_t = b \times y_{t-1} + \varepsilon_t, \quad (2.1)$$

$$\Delta y_t = b_0 + b \times y_{t-1} + \varepsilon_t, \quad (2.2)$$

$$\Delta y_t = b_0 + b_1 \times t + b \times y_{t-1} + \varepsilon_t, \quad (2.3)$$

де Δ – оператор першого диференціювання,

y_t – змінна;

b_0 – константа;

ε_t – похибка.

Перший випадок характеризується відсутністю константи і тренду. Другий наявністю константи, проте відсутністю тренду. Третій наявністю як константи, так і лінійного тренду.

Кожний з трьох видів тестової регресії має власні критичні значення DF-тесту, дані значення можемо спостерігати у таблиці 2.1.

Таблиця 2.1 – Критичні значення тесту Діккі-Фуллера

Види тестової регресії	I		II		III	
Об’єм вибірки	1%	5%	1%	5%	1%	5%
N=25	-2,66	-1,95	-3,75	-3,00	-4,38	-3,60
N=50	-2,62	-1,95	-3,58	-2,93	-4,15	-3,50
N=100	-2,60	-1,95	-3,51	-2,89	-4,04	-3,45
N=500	-2,58	-1,95	-3,44	-2,87	-3,98	-3,42

2.2 Методи заповнення пропусків даних

2.2.1 Метод випадкового лісу

Усі порожні значення вибірки проходять ініціалізацію шляхом зміни їх середнім значенням за деякою ознакою. У якості алгоритму для уточнення порожніх значень обирається випадковий ліс.

Спочатку на об'єктах що не містять пропуски проходить навчання. Цей процес відбувається для кожної ознаки, у якій необхідно виконати заміну пропусків. Далі відбувається заміна значень даної ознаки для об'єктів, що залишилися, за допомогою алгоритму. Алгоритм працює або до збіжності або до деякого заздалегідь встановленого числа ітерацій.

2.2.2 Метод k-найближчих сусідів

Головною ідеєю даного методу виступає припущення, щодо того, що об’єктам що близько розташовані одне від одного властиві подібні ознаки. Тобто, якщо допустити, що існує певний об’єкт x, у якого присутні пропуски значень, то можна знайти об’єкти $x_1, \dots x_k$, та на основі їх значень визначити пропущені значення x. Цей процес відбувається за наступною формулою:

$$x = \frac{\sum_{k=1}^{K} x_k}{K}, \tag{2.4}$$

де x – об’єкт з пропущеним значенням;

x_k – об’єкт без пропусків значень.

2.3 Метод фільтрації даних

Фільтр- це певний алгоритм обробки вхідних даних, з його допомогою данні позбуваються шуму, а також зайвої інформації.

Експоненціальний фільтр – один із найпопулярніших фільтрів першого порядку. Крім того його ще можна назвати найпростішим фільтром першого порядку. Його принцип роботи показано на рисунку 2.3.

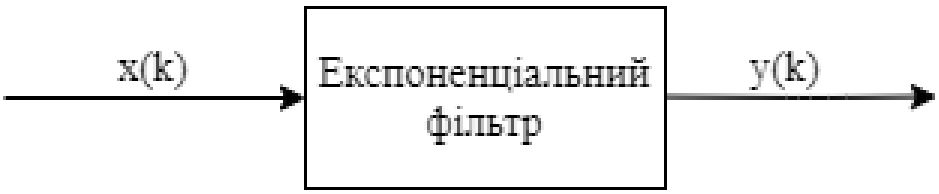


Рисунок 2.3 – Принцип роботи експоненціального фільтру

Аналітичне представлення експоненціального фільтру:

$$y_k = a \times x_k + (1 - a) \times y_{k-1} \text{ або } y_k = y_{k-1} + a \times (x_k - y_{k-1}), \quad (2.5)$$

$$a = 1 - e^{\frac{-\Delta T}{\tau}}, \quad (2.6)$$

де x_k – вхідні значення у момент k;

y_k- вихідні значення у момент k;. що пройшли фільтрацію;

a − коефіцієнт фільтрації;

ΔT - інтервал часу вибірки;

τ – константа часу фільтра.

При збільшенні коефіцієнта a, степінь фільтрації зменшується, і навпаки при зменшенні a − збільшується. У випадку коли a = 0, на виході можемо отримати максимальну фільтрацію даних, а у свою чергу, при a = 1 − мінімальну. Найчастіше використовується значення не більше за 0,5.

У даного фільтра є низка переваг:

- доволі легкий у застосуванні;
- пригнічує різкі скачки вимірів;
- демонструє доволі точні результати прогнозів.

Недоліком експоненціального фільтру є значна затримка.

2.4 Моделі для прогнозування

2.4.1 Авторегресійна модель

Авторегресійна модель позначається як AR(p) та являє собою стаціонарний процес вигляду:

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_p y_{t-p} + \varepsilon_t, \quad (2.7)$$

де ε_t- білий шум, або іншими словами послідовність незалежних та однаково розподілених випадкових величин;

a_0-константа;

p - порядок авторегресії (кількість лагів авторегресії);

q - порядок ковзного середнього (кількість лагів КС);

$a_1, a_2, \dots, a_p$- авторегресійні параметри моделі.

Для збереження даною моделлю стаціонарності, існують наступні обмеження відносно її параметрів. Як приклад, у випадку AR(1) таким обмеженням є:

$$|a_1| < 1, \quad (2.8)$$

де a_1- авторегресійні параметри моделі.

2.4.2 Авторегресія з ковзним середнім

Модель авторегресії з ковзним середнім ARMA(p,q) компонує у собі дві більш прості моделі – модель авторегресії (AR) та модель ковзного середнього (MA), та та подається у вигляді наступного стаціонарного процесу:

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_p y_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + b_2 \varepsilon_{t-2} + \cdots + \\ \ldots + + b_q \varepsilon_{t-q}, \quad (2.9)$$

де ε_t- білий шум, а якщо точніше, то – послідовність незалежних та однаково розподілених випадкових величин;

a_0-константа;

p – порядок авторегресії;

q – порядок КС;

$a_1, a_2, \ldots, a_p$- авторегресійні коефіцієнти моделі;

$b_1, b_2, \ldots, b_q$ – авторегресійні коефіцієнти ковзного середнього.

Сума p+q є мінімально можливою. За допомогою даної моделі, ми можемо точно й компактно описати будь-який стаціонарний процес.

Крім того модель ARMA може бути подана у альтернативному вигляді, за допомогою оператора затримки L(p,q):

$$\left(1 - \sum_{i=1}^{p} a_i L_i\right) y_t = \left(1 - \sum_{j=1}^{q} b_j L_j\right) \varepsilon_t, \quad (2.10)$$

$$\text{або } A y_t = B \varepsilon_t, \quad (2.11)$$

де B та A – поліноми виду:

$$B = 1 - \sum_{j=1}^{q} b_j L_j, \tag{2.12}$$

$$A = 1 - \sum_{i=1}^{p} a_i L_i, \tag{2.13}$$

де $a_1, a_2, \dots, a_p$- авторегресійні коефіцієнти моделі;

$b_1, b_2, \dots, b_q$ – авторегресійні коефіцієнти ковзного середнього.

2.4.3 LSTM (Long short-term memory)

LSTM (long short-term memory, дослівно (довга короткострокова пам'ять) - тип рекуррентной нейронної мережі, здатний навчатися довгостроковим залежностям. LSTM були представлені в роботі [Hochreiter & Schmidhuber (1997)], згодом вдосконалені і популяризували іншими дослідниками, добре справляються з багатьма завданнями і до сих пір широко застосовуються.

LSTM спеціально розроблені для усунення проблеми довгострокової залежності. Їх спеціалізація - запам'ятовування інформації протягом тривалих періодів часу, тому їх практично не потрібно навчати.

Всі рекурентні нейронні мережі мають форму ланцюжка повторюваних модулів нейронної мережі. У стандартних РНМ цей повторюваний модуль має просту структуру, наприклад, один шар tanh, як показано на рисунку 2.4.

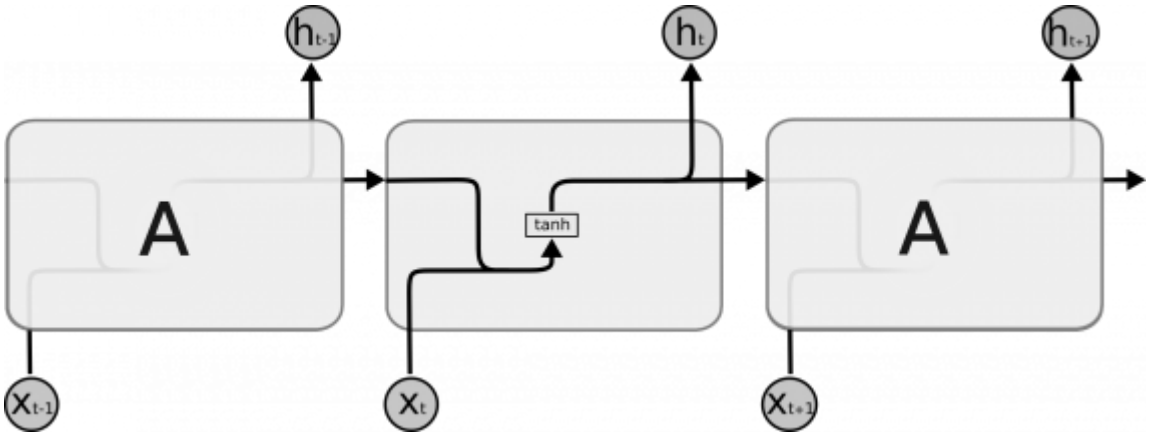


Рисунок 2.4 – Структурна схема стандартної РНМ

У свою чергу повторюваний модуль LSTM складається з чотирьох взаємодіючих шарів, що можна побачити на рисунку 2.5.

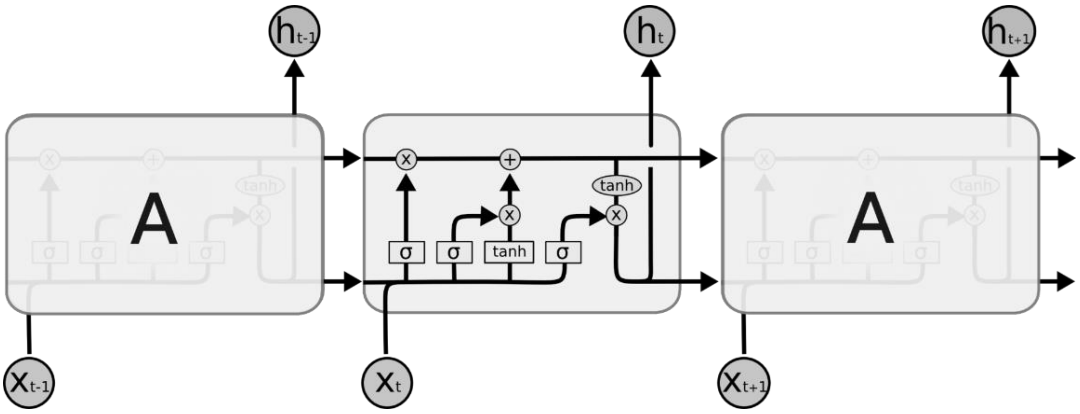


Рисунок 2.5 – Структурна схема мережі LSTM

Ключовим моментом роботи LSTM є стан комірки, який частково нагадує конвеєрну стрічку. Вона проходить через весь ланцюжок, піддаючись незначним лінійним перетворенням (рисунок 2.6).

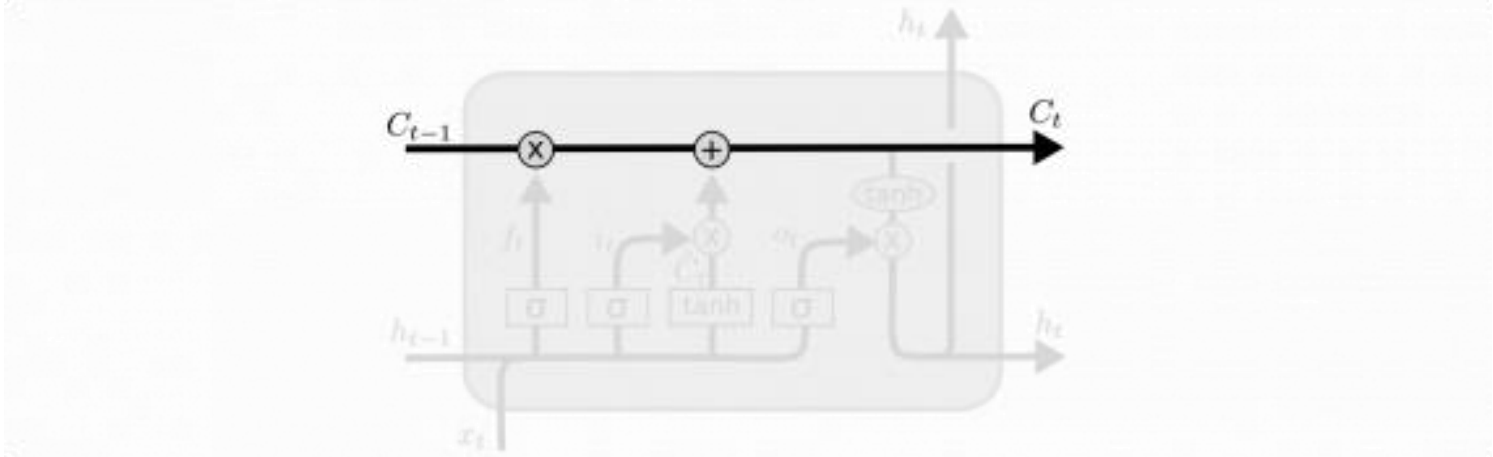


Рисунок 2.6 - Структурна схема стану комірок мережі LSTM

У LSTM є можливість зменшувати або збільшувати кількість інформації в стані комірки, в залежності від потреб. Для цього використовуються ретельно налаштовані структури, так звані гейти. Гейт - це «ворота» або «фільтри», що мають здатність пропускаюти або затримувати інформацію. Гейти складаються з сигмоїдного шару нейронної мережі і операції точкового множення.

На виході сигмоїдного шару можемо отримати числа від нуля до одиниці, які визначають, скільки відсотків кожної одиниці інформації пропустити далі. Значення «0» означає «не пропустити нічого», значення «1» - «пропустити все».

Зазвичай, у стандартному вигляді, модель LSTM складається з чотирьох основних шарів:

1) вхідний шар;
2) шар забуття;
3) шар збереження;
4) вихідний шар.

Структура шару забуття наведена на рисунку 2.7.

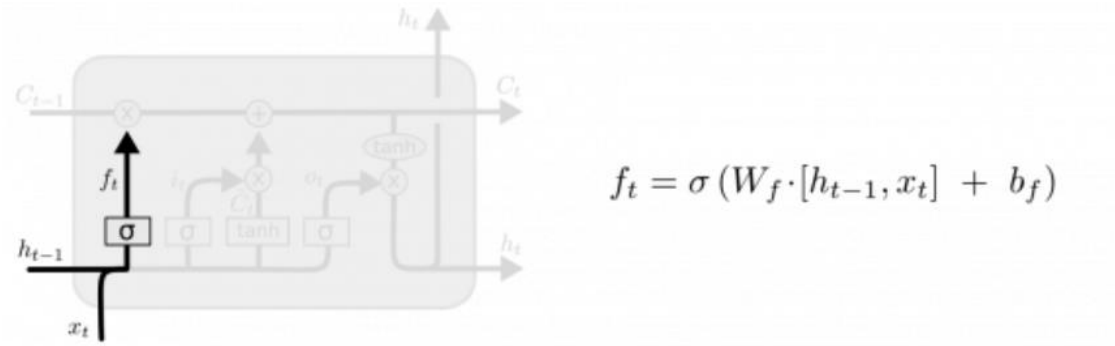
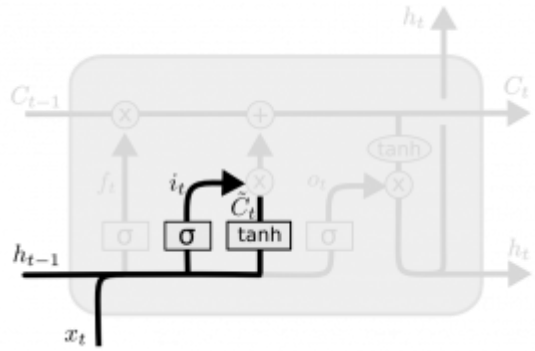


Рисунок 2.7 - Структурна схема шара забуття моделі LSTM

LSTM потрібно вирішити, яку інформацію ми збираємося вилучити зі стану комірки. Це рішення приймається сигмоїдним шаром, так званим «шаром забуття». Він отримує на вхід h_{t-1} і x_t і видає число від 0 до 1 для кожного номера в стані комірки C. 1 означає «повністю зберегти», а 0 - «повністю видалити».

На наступному кроці потрібно вирішити, яку нову інформацію зберегти в стані комірки. Розбивши процес на дві частини, бачимо, що спочатку сигмоїдний шар «шаром гейта входу», вирішує, які значення потрібно оновити. Потім шар tanh створює вектор нових значень-кандидатів C, які додаються в уже новий стан. На наступному кроці ми об'єднаємо ці два значення для поновлення стану. Даний процес схематично показаний на рисунку 2.8.

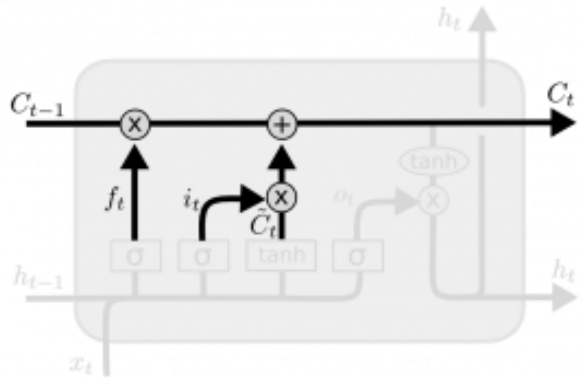


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Рисунок 2.8 - Структурна схема шара збереження моделі LSTM

Вхідний фільтр (input layer gate) визначає, чи зберігати нові дані до стану комірки. У вхідному зфільтрі значення, яке потрібно оновити, визначається сигмовидною функцією, а вектор, який потрібно додати до стану комірки, генерується функцією tanh. Стан комірки оновлює попередній стан до нового стану (рисунок 2.9).



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Рисунок 2.9 - Процес оновлення стану комірки LSTM

Наступним кроком виступає оновлення попереднього стану комірки (рисунок 2.10) для отримання нового стану C. Оновлення відбувається наступним чином: множення старого стану на f, втрачаючи інформацію, яку вирішили забути; потім додаємо $i_t \times \tilde{C}_t$. Це нові значення кандидатів, масштабовані в залежності від того, яким чином ми вирішимо оновити кожне значення стану.

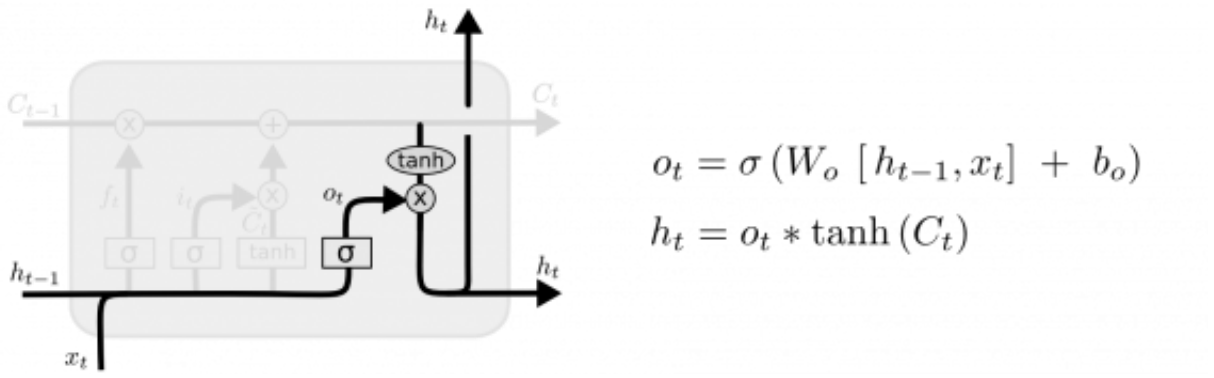


Рисунок 2.10 - Процес оновлення стану комірки LSTM

Далі, переходимо до шару що відповідає за кінцевий результат. У нашому випадку результатом буде відфільтрований стан комірки. Спочатку запускаємо сигмоїдний шар, який вирішує, які частини стану комірки подавати на вихід. Потім пропускаємо стан комірки через tanh (щоб нормувати всі значення в інтервалі [-1, 1]) і множимо його на вихідний сигнал сигмоїдного гейту.

2.4.4 Множинна регресія

Лінійна (OLS) регресія робить порівняння відповіді залежної змінної з урахуванням зміни деякої пояснювальної змінної. Проте лише у небагатьох випадках трапляється, коли залежна змінна пов’язана тільки з однією змінною. Тому у більшості випадків доводиться використовувати множину регресію, яка

пов’язує залежну змінну з декількома незалежними. Множинні регресії бувають двох типів: лінійні та нелінійні.

Множинна регресія працює на гіпотезі існування лінійної залежності як залежної, так і незалежних змінних. Крім того, вона не допускає значної кореляції між незалежними змінними. Даний вид регресії в загальному вигляді можна розглядати як численні регулярні лінійні регресійні моделі. Це пояснюється тим, що відбувається просте співвідношення між ознаками для деякоїзаданої кількості ознак. Припускається, що між залежною змінною та незалежною змінною є лінійна залежність, а також що змінні - це постійні значення, а не дискретні.

Процес множинної регресії демонструє кореляційний зв’язок між деякою залежною змінною (y) та декількома незалежними змінними ($x_1, x_2, \dots, x_p$), тобто

$$y = f(x_1, x_2, \dots, x_p), \tag{2.14}$$

Модель множинної лінійної регресії представляє собою стаціонарний процес вигляду:

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_px_p + \varepsilon, \tag{2.15}$$

де a_0-константа;

$a_1, a_2, \dots, a_p$- параметри моделі.

Альтернативний запис:

$$Y = Xa + \varepsilon, \tag{2.16}$$

де X– незалежні змінні;

Y-значення залежних змінних,

a- значення параметрів регресії та вільного члена a_0,

ε - вектор значень випадкових залишків.

2.4.5 Сезонна модель ARIMA (SARIMA)

Модель ARIMA(p,d,q) - представляє собою інтегровану модель авторегресії з ковзним середнім ARMA(p,q). Дана модель має вигляд:

$$\Delta^d y_t = a_0 + a_1 \Delta y_{t-1} + a_2 \Delta y_{t-2} + \cdots + a_p \Delta y_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + \\ + + b_2 \varepsilon_{t-2} + \cdots + b_q \varepsilon_{t-q}, \quad (2.17)$$

де p - порядок авторегресії (кількість лагів авторегресії);

q - порядок ковзного середнього (кількість лагів КС);

a_0-константа;

$a_1, a_2, \dots, a_p$- параметри моделі;

$b_1, b_2, \dots, b_q$- параметри моделі;

ε_t- білий шум;

Δ^d- оператор різниці порядку d.

У випадку, коли d=0, з моделі ARIMA отримаємо звичайну модель авторегресії з ковзним середнім ARMA.

Модель SARIMA(p,d,q)(P,D,Q)m ж є розширеною версією моделі ARIMA(p,d,q), яка дозволяє не лише обробляти дані з трендом, а й враховувати сезонний компонент ряду.

Перша частина параметрів моделі SARIMA(p,d,q)(P,D,Q) повністю аналогічна моделі ARIMA(p,d,q), тут p - порядок авторегресії, q - порядок ковзного середнього, d – порядок різниці. Проте, у даній моделі ще додаються наступні параметри: P – сезонний порядок авторегресії, Q – сезонний порядок ковзного

середнього, D – порядок сезонної різниці та m – кількість інтервалів одного сезонного періоду.

2.4.6 Suport Vector Regression

SVM (від англ. Support Vector Machines) - це лінійний алгоритм, що доволі часто використовують в задачах класифікації та регресії. У SVM є широке застосування на практиці за допомогою нього можна вирішити як лінійні так і нелінійні задачі. Тренування первинної ОВР має на увазі вирішення наступного виразу:

$$min\frac{1}{2}\|w\|^2, \tag{2.18}$$

2.4.7 XGBoost

Це метод машинного навчання для задач класифікації та регресії, який будує модель передбачення в формі ансамблю слабких моделей, в загальному вигляді як дерева рішень.

Основоположною частиною моделі XGBoost виступає алгоритм градієнтного посилення (Gradient boosting) дерев рішень. Даний алгоритм є технікою машинного навчання, що використовується для задач класифікації та регресії, а також побудови моделей прогнозування у вигляді ансамблю слабких пророчих моделей, найчастіше у формі дерев рішень.

Навчання даного ансамблю відбувається послідовно, кожну ітерацію виконується обчислення похибки прогнозу вже навченого ансамблю порівняно з навчальною вибіркою. Кожна модель, що буде далі додаватись до ансамбля буде передбачати ці відхилення. Тож додавши дані прогнозування нового дерева до прогнозувань ансамблю, що вже є навченим, відбувається зменшення середнього відхилення моделі. Нові дерева рішень додаються до того моменту, поки відбувається зменшення помилки, або поки не буде виконана хоча б одна з умов зупинки.

Найголовніша перевага XGBoost — його масштабованість у всіх сценаріях. Система XGBoost виконує обчислення в рази швидше, ніж існуючі відомі аналоги. Її масштабованість перш за все пояснюється наступними алгоритмічними оптимізаціями:

– можливість обробки вагів екземплярів у приблизному вивченні дерев, як наслідок теоретично обгрунтованої та зваженої процедури пошуку похибки;
– навчення стає на порядок швидшим, а як наслідок модель навчається швидше за рахунок паралельних та розподілених обчислень.
– Оптимізація градієнтного бустингу відбувається за наступною формулою:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t), \quad (2.19)$$

де l – функція втрат;

y_i – фактичне (реальне) значення часового ряду;

$\hat{y}_i$ – значення прогнозу.

x_i – набір ознак і-го елемента навчальної вибірки;

f_t – функція (дерево), яку ми хочемо навчити на кроці t;

$f_t(x_i)$ – передбачення на і-му елементі навчальної вибірки;

$\Omega\ (f)$ – регулювання функції f.

2.5 Критерії адекватності математичних моделей і якості прогнозів

2.5.1 Критерії адекватності математичних моделей

Математична модель може вважатись адекватною лише у тому випадку, якщо :

- відображає найхарактернішу взаємодію між усіма змінними процесу;
- враховує усі можливі сигнали;
- враховує вплив шумів та зовнішніх збурень;
- враховує обмеження на значення змінних.

AIC (інформаційний критерій Акайке) – критерій, що використовується для вибору найкращої статистичної моделі.

$$\mathrm{AIC} = 2k - 2\ln(L), \tag{2.20}$$

де k – кількість параметрів оцінюваної моделі;

L – максимальне значення ф-ції правдоподібності оцінюваної моделі.

$R^2 -$ коефіцієнт детермінації, визначається як рівень залежності варіації залежної змінної від незалежних.

$$R^2 = 1 - \frac{V(y|x)}{V(y)} = 1 - \frac{\sigma^2}{\sigma_y^2}, \tag{2.21}$$

де V(y | x) - умовна дисперсія залежної змінної (дисперсія помилки моделі);
V(y) - дисперсія ВВ y.

Чим більше значення коефіцієнта детермінації тим більша відповідність моделі даним, отже й краща модель. Якщо $R^2 = 1$, тоді між змінними функціональна залежність. На практиці, прийнятними є моделі зі значенням коефіцієнта детермінації більще 0.5.

Критерій Дарбіна-Уотсона використовується для визначення наявності автокореляції залишків першого порядку регресійної моделі.

$$DW = \frac{\sum_{t=2}^{n}(\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^{n}\varepsilon_t{}^2} \approx 2(1 - p_1) \qquad (2.22)$$

де $p_1 -$ коефіцієнт автокореляції першого порядку.

Якщо $DW = 2$ – це свідчить про те, що автокореляція відсутня.

У випадку, якщо критерій DW прямує до 0, можемо робити висновок про наявність позитивної автокореляції, а якщо DW прямує до 4 – про наявність негативної.

2.5.2 Критерії якості прогнозів

Під час проведення експериментів якість одержаного прогнозу будемо визначати за допомогою наступних критеріїв:

MSE (Mean Square Error) – середньоквадратична похибка моделі:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (2.23)$$

де y_i – реальне значення часового ряду;

$\hat{y}_i$ – значення, яке було зпрогнозоване.

RMSE (Root Mean Square Error) корінь з середньоквадратичної похибки:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{2.24}$$

де y_i – реальне значення часового ряду;

$\hat{y}_i$ – значення, яке було зпрогнозоване.

MAE (Mean absolute error) – середня абсолютна похибка:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{2.25}$$

де y_i – реальне значення часового ряду;

$\hat{y}_i$ – значення, яке було зпрогнозоване.

MAPE (Mean absolute percentage error)- середня абсолютна похибка в процентах:

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \times 100\%, \tag{2.26}$$

де y_i – фактичне значення часового ряду;

$\hat{y}_i$ – значення, яке було зпрогнозоване.

Theil – коефіцієнт Тейла:

$$\mathrm{Theil} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{T}(y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{T}(y_i)^2} + \sqrt{\frac{1}{N}\sum_{i=1}^{T}(\hat{y}_i)^2}} \tag{2.27}$$

де y_i –реальне значення ЧР;

$\hat{y}_i$ – зпрогнозоване значення.

Якщо $\mathrm{Theil} = 1$ – це свідчення того, що побудована модель не підходить для прогнозування. І навпаки, якщо коефіцієнт $\mathrm{Theil} = 0$, можемо зробити висновок, що побудована модель — ідеальна.

2.6 Висновки до розділу 2

У даному розділі були наведені основні поняття та характеристики часового ряду. Наведені статистичні тести, що дозволяють провести перевірку часового ряду на стаціонарність. Також були розглянута теорія що стосувалася методів заповнення пропусків – метод випадкового лісу, метод k-найближчих сусідів, та підхід до фільтрації даних у вигляді експоненціального фільтру. Також було наведено математичні моделі для опису та прогнозування ЧР. Крім того було наведено критерії, що дозволяють оцінити адекватність відповідної математичної моделі та якість прогнозу.

РОЗДІЛ 3 ПОБУДОВА МОДЕЛЕЙ ТА ПРОГНОЗІВ ДЛЯ ОБРАНИХ ПРОЦЕСІВ

3.1 Аналіз вхідних даних

У якості практичної частини моєї магістерської дисертації виступила розробка та реалізація програмного продукту для побудови та аналізу вищезазначених моделей прогнозування часових рядів та для аналізу та коригування вхідних даних.

У цьому розділі усі експерименти будемо проводити зі статистичними даними комерційного банку «COTA Bank (Cota Commercial Bank) Taiwan», що були частково отримані безпосередньо зі статистичних звітностей з офіційного сайту банку а також були розширені та доповнені з фінансових платформ World Bank Open Data та Global Financial Data. Дані подаються у вигляді щотижневих фінансових показників та охоплюють часовий проміжок з 2015-го року по 2021-й рік. Для подальшої роботи було обрано наступні фінансові показники:

1. Рентабельність активів (return on assets, ROA) – фінансовий коефіцієнт, що характеризує віддачу від використання всіх активів організації (рисунок 3.1 – 3.2).
2. Коефіцієнт швидкої ліквідності (Quick ratio) – показник здатності компанії погашати поточні зобов'язання за рахунок оборотних активів (рисунок 3.3 – 3.4).
3. Коефіцієнт кредитного ризику – показник можливості збитку, що виникає внаслідок невиконання позичальником кредиту або виконання договірних зобов’язань(рисунок 3.5 – 3.6).
4. Активи банку (Asset) – фінансові ресурси банку до яких відносяться як власні так і позикові кошти (рисунок 3.7 – 3.8).
5. Показник співвідношення позик до активів (Loan-to-Assets Ratio) банку(рисунок 3.9 – 3.10).
6. Коефіцієнт покриття ліквідністю (LCR) (рисунок 3.11 – 3.12).

7. Фінансовий важіль (Leverage Ratio)- показник залежності прибутку від розміру використання позики(рисунок 3.13 – 3.14).

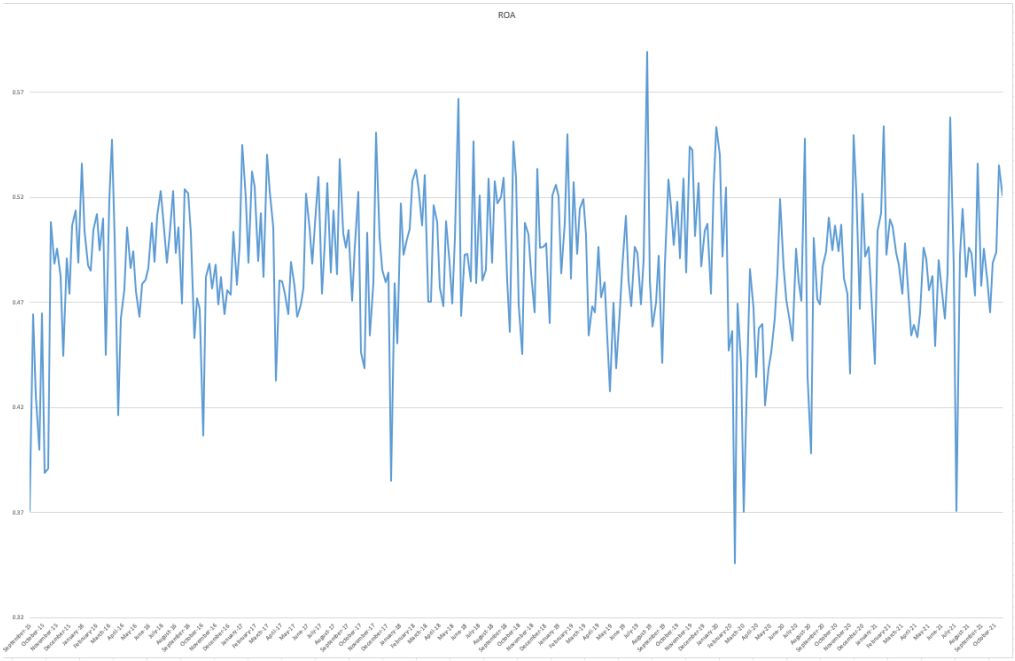


Рисунок 3.1 – Графік коефіцієнта рентабельності активів (ROA) за досліджуваний період

```
Name: ROA, dtype: float64
count    mean     std      min      25%      50%      75%      max
321.00   0.48     0.03     0.34     0.47     0.48     0.50     0.58
```

Рисунок 3.2 – Коротка інформація про часовий ряд (ROA)

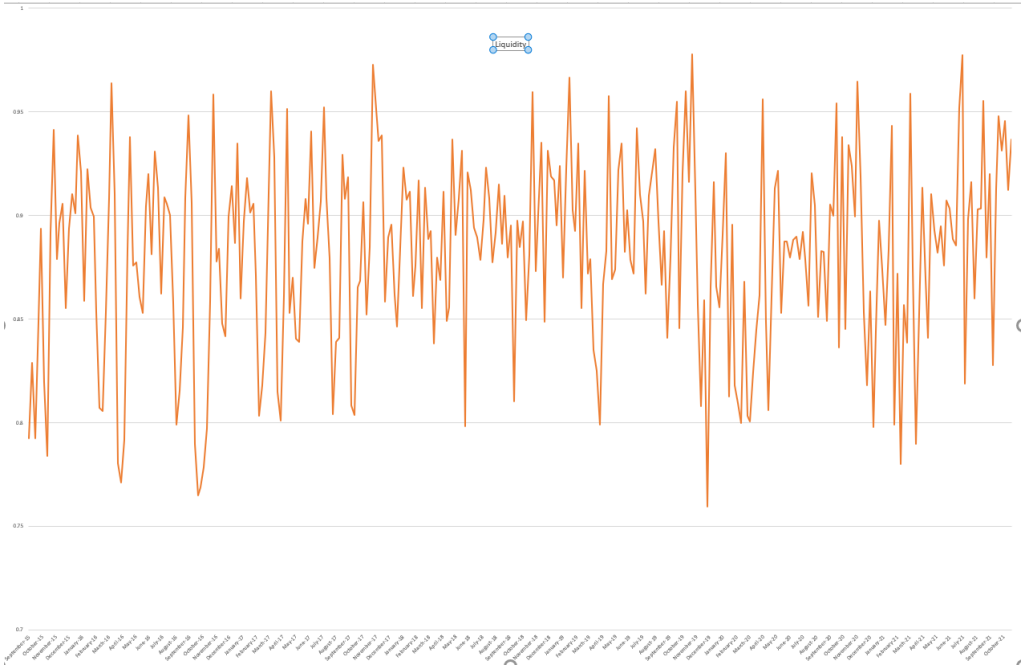


Рисунок 3.3 – Графік коефіцієнта швидкої ліквідності за досліджуваний період

```
Name: Liquidity, dtype: float64
count    mean     std      min      25%      50%      75%      max
321.00   0.88     0.04     0.75     0.85     0.88     0.91     0.97
```

Рисунок 3.4 – Коротка інформація про часовий ряд (Quick ratio)

Рисунок 3.5 – Графік коефіцієнта кредитного ризику за досліджуваний період

```
Name: Credit_risk, dtype: float64
count    mean     std      min      25%      50%      75%      max
321.00   0.222    0.001    0.210    0.222    0.222    0.222    0.232
```

Рисунок 3.6 – Коротка інформація про часовий ряд (Credit risk)

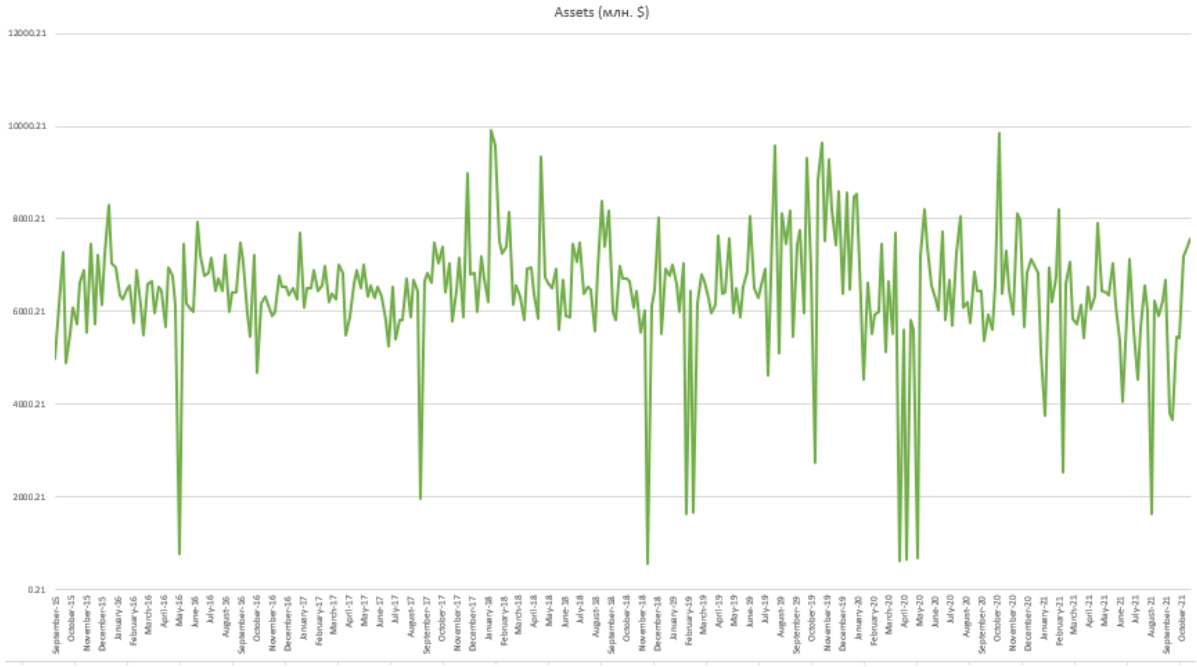


Рисунок 3.7 – Графік зміни розміру активів банку за досліджуваний період

```
Name: Assets (million. $), dtype: float64
count    mean     std      min      25%      50%      75%      max
321.0    6423.7   1346.7   566.0    5980.0   6480.0   7020.0   9910.0
```

Рисунок 3.8 – Коротка інформація про часовий ряд (Asset)

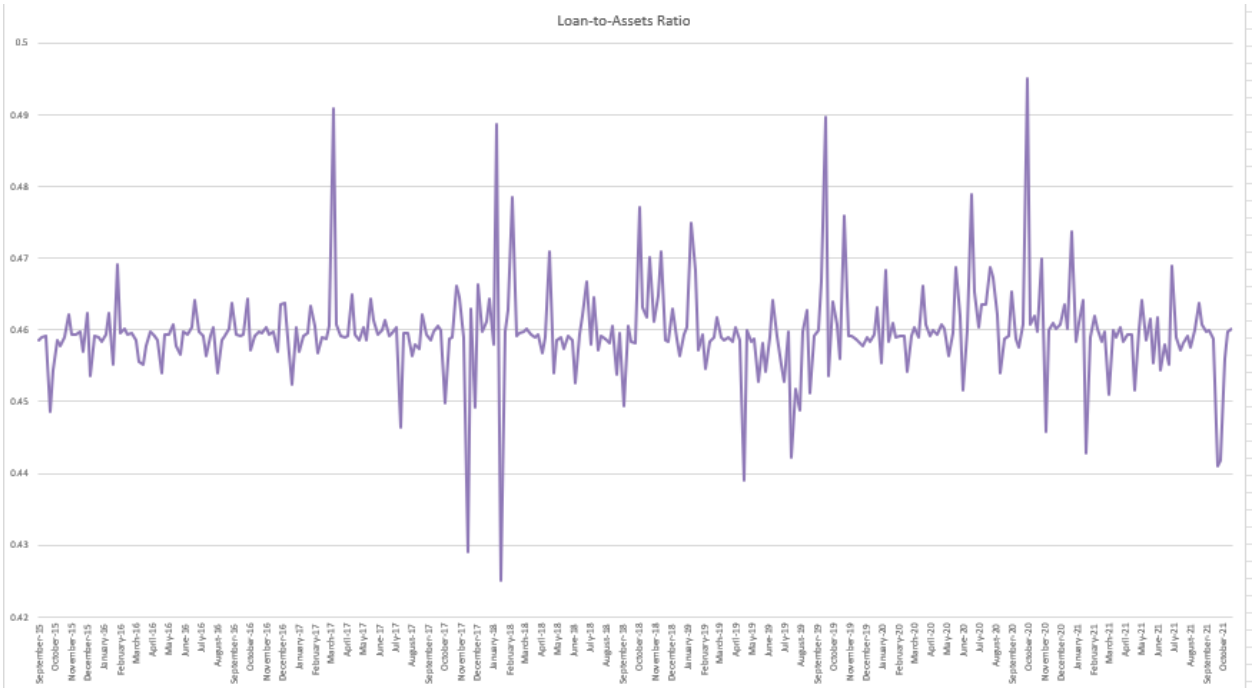


Рисунок 3.9 – Графік співвідношення позик до активів за досліджуваний період

count	mean	std	min	25%	50%	75%	max
321.000	0.459	0.006	0.425	0.458	0.459	0.461	0.495

Рисунок 3.10 – Коротка інформація про часовий ряд (Loan-to-Assets Ratio)

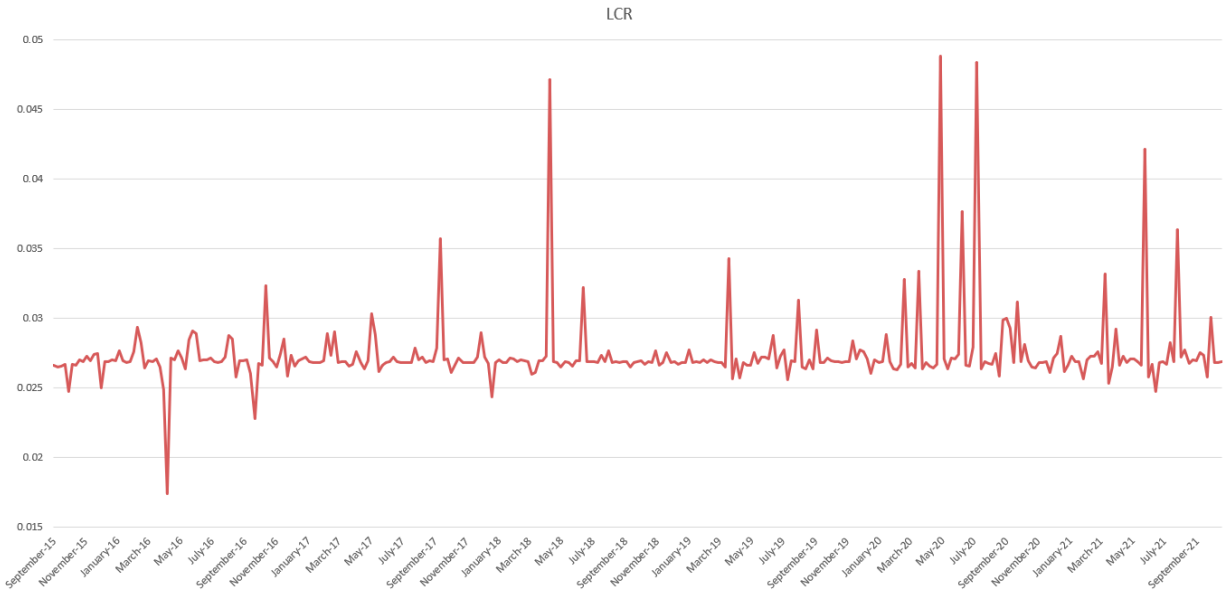


Рисунок 3.11 – Графік зміни коефіцієнта покриття ліквідністю за досліджуваний період

count	mean	std	min	25%	50%	75%	max
321.000	0.027	0.002	0.017	0.026	0.026	0.027	0.048

Рисунок 3.12 – Коротка інформація про часовий ряд (LCR)

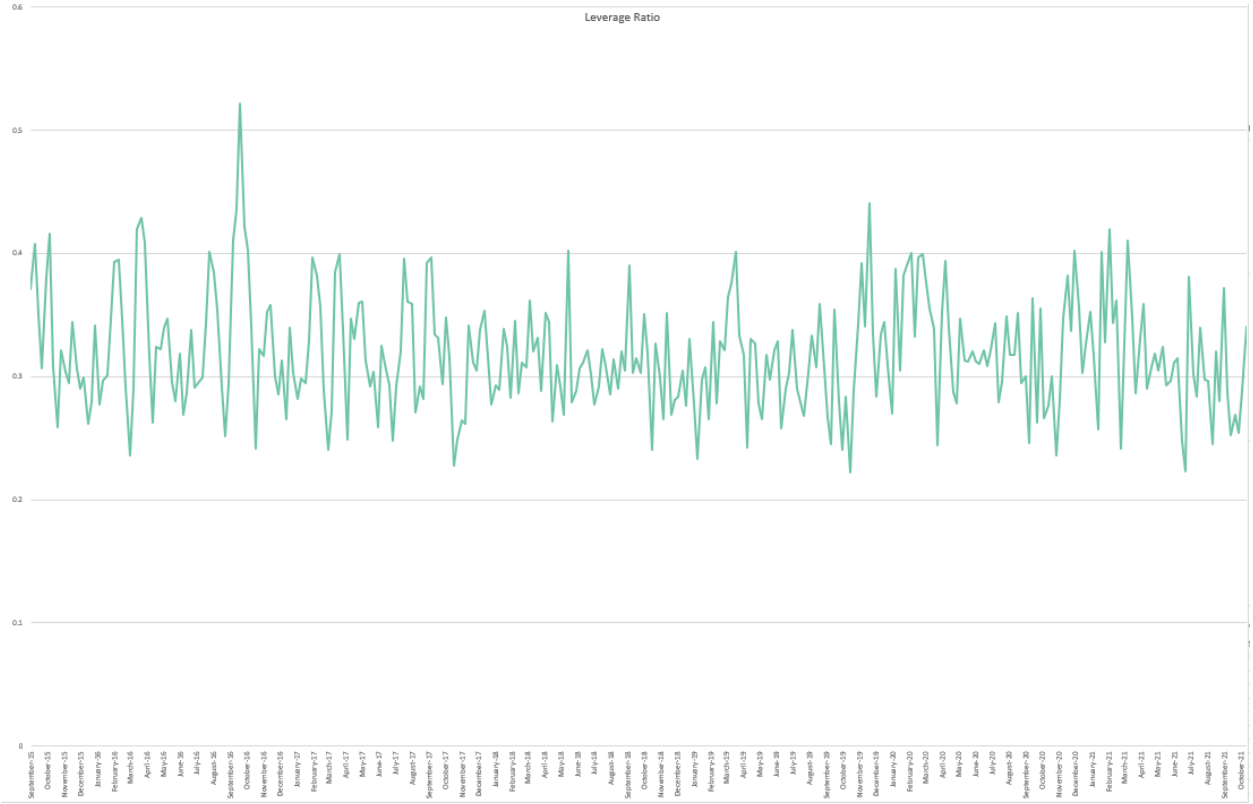


Рисунок 3.13 – Графік зміни коефіцієнту використання позикових засобів за досліджуваний період

count	mean	std	min	25%	50%	75%	max
321.00	0.31	0.04	0.22	0.28	0.31	0.34	0.52

Рисунок 3.14 – Коротка інформація про часовий ряд (Leverage Ratio)

3.2 Аналіз та прогнозування досліджуваних часових рядів

Найперше що потрібно виконати — знайти та замінити усі пропуски даних у вибірці. Найзручнішим рішенням для подальшої програмної реалізації буде заміна пропусків даних спеціальним значенням, у нашому випадку будемо використовувати значення «-1» (рисунок 3.15).

	A	B	C	D	E	F	G	H	I	J
130	Oct-05	0.524204	0.855096	0.512019	8160000000	8160	0.222552	-1	0.027011	0.344904314
131	Nov-05	0.506508	0.913233	0.515675	6160000000	6160	-1	0.459506	0.026963	0.28676724
132	Dec-05	0.530737	0.888624	0.512925	6550000000	6550	0.222401	0.459839	0.026906	0.311376485
133	Jan-06	0.470287	0.892554	0.518268	6320000000	6320	0.219651	0.460163	0.025964	0.307446262
134	Feb-06	0.470141	0.837954	0.503874	5820000000	5820	0.220776	0.459465	0.026125	0.362046135
135	Mar-06	0.516307	0.879826	0.493754	6920000000	6920	0.22242	0.458934	0.026926	0.320174139
136	Apr-06	0.508312	0.868882	0.50052	6950000000	6950	0.222472	0.459342	0.026963	0.3311183
137	May-06	0.47721	0.91151	0.502857	6380000000	6380	0.222722	0.45686	0.027243	0.288490492
138	Jun-06	0.468142	0.848838	0.495163	5840000000	5840	-1	0.45899	0.047109	0.351162439
139	Jul-06	0.508556	0.855489	0.487057	9360000000	9360	-1	0.470974	0.026865	0.344511292
140	Aug-06	0.488666	-1	0.493506	6730000000	6730	-1	0.453994	0.026807	0.263216132
141	Sep-06	0.469556	-1	0.492054	6630000000	6630	0.221732	0.458605	0.026482	0.309562536
142	Oct-06	0.500122	0.908063	0.485305	6500000000	6500	0.222407	0.459011	0.026885	0.291936995
143	Nov-06	0.567055	0.931161	0.48975	6930000000	6930	0.222245	0.457308	0.026791	0.268839375
144	Dec-06	0.463365	0.798228	0.512941	5620000000	5620	0.221864	0.459209	0.026534	0.401771624
145	Jan-07	0.492858	0.920882	0.498068	6690000000	6690	0.222443	0.458576	0.026912	0.279118421
146	Feb-07	0.493248	0.912235	0.496505	5920000000	5920	-1	0.452679	0.026918	0.287764912
147	Mar-07	0.479891	0.893824	0.495528	5890000000	5890	-1	0.459394	-1	0.306176497
148	Apr-07	0.546678	0.889258	0.490837	7450000000	7450	0.222331	0.462054	-1	0.310741603
149	May-07	0.479355	0.878526	0.507589	7080000000	7080	0.222391	0.466679	0.026888	0.321474136
150	Jun-07	0.520743	0.897723	0.499119	7500000000	7500	0.222261	0.458966	0.026878	0.302276506

Рисунок 3.15 – Частина вибірки із заміненими спеціальним значенням пропусками

Пропуски будемо заповнювати за допомогою методу випадкового лісу. Після його застосування, отримуємо наступні результати (рисунок 3.16).

	A	B	C	D	E	F	G	H	I	J
130	Oct-05	0.524204	0.855096	0.512019	8160000000	8160	0.222552	0.459157	0.027011	0.344904314
131	Nov-05	0.506508	0.913233	0.515675	6160000000	6160	0.222493	0.459506	0.026963	0.28676724
132	Dec-05	0.530737	0.888624	0.512925	6550000000	6550	0.222401	0.459839	0.026906	0.311376485
133	Jan-06	0.470287	0.892554	0.518268	6320000000	6320	0.219651	0.460163	0.025964	0.307446262
134	Feb-06	0.470141	0.837954	0.503874	5820000000	5820	0.220776	0.459465	0.026125	0.362046135
135	Mar-06	0.516307	0.879826	0.493754	6920000000	6920	0.22242	0.458934	0.026926	0.320174139
136	Apr-06	0.508312	0.868882	0.50052	6950000000	6950	0.222472	0.459342	0.026963	0.3311183
137	May-06	0.47721	0.91151	0.502857	6380000000	6380	0.222722	0.45686	0.027243	0.288490492
138	Jun-06	0.468142	0.848838	0.495163	5840000000	5840	0.223978	0.45899	0.047109	0.351162439
139	Jul-06	0.508556	0.855489	0.487057	9360000000	9360	0.222343	0.470974	0.026865	0.344511292
140	Aug-06	0.488666	0.936784	0.493506	6730000000	6730	0.222268	0.453994	0.026807	0.263216132
141	Sep-06	0.469556	0.890437	0.492054	6630000000	6630	0.221732	0.458605	0.026482	0.309562536
142	Oct-06	0.500122	0.908063	0.485305	6500000000	6500	0.222407	0.459011	0.026885	0.291936995
143	Nov-06	0.567055	0.931161	0.48975	6930000000	6930	0.222245	0.457308	0.026791	0.268839375
144	Dec-06	0.463365	0.798228	0.512941	5620000000	5620	0.221864	0.459209	0.026534	0.401771624
145	Jan-07	0.492858	0.920882	0.498068	6690000000	6690	0.222443	0.458576	0.026912	0.279118421
146	Feb-07	0.493248	0.912235	0.496505	5920000000	5920	0.222441	0.452679	0.026918	0.287764912
147	Mar-07	0.479891	0.893824	0.495528	5890000000	5890	0.223696	0.459394	0.032231	0.306176497
148	Apr-07	0.546678	0.889258	0.490837	7450000000	7450	0.222331	0.462054	0.026848	0.310741603
149	May-07	0.479355	0.878526	0.507589	7080000000	7080	0.222391	0.466679	0.026888	0.321474136
150	Jun-07	0.520743	0.897723	0.499119	7500000000	7500	0.222261	0.458966	0.026878	0.302276506

Рисунок 3.16 – Частина вибірки із заповненими пропусками методом випадкового лісу

Далі будемо застосовувати процедуру фільтрації експоненціальним фільтром, задля позбавлення шуму та зайвої інформації (рисунки 3.17-3.19). Коефіцієнт фільтрації θ = 0.3.

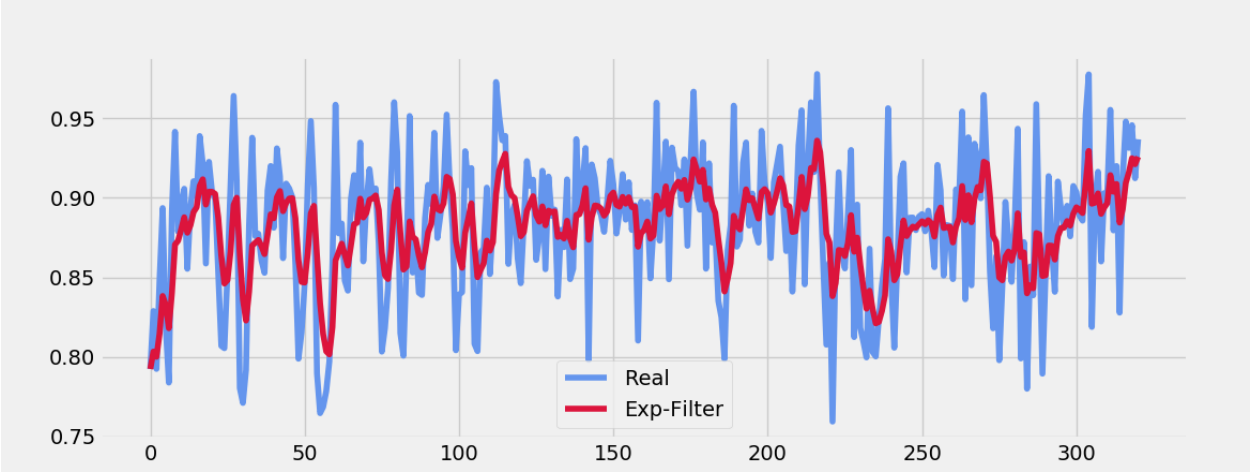


Рисунок 3.17 – Графік даних коефіцієнта швидкої ліквідності (Quick ratio) з використанням експоненціального фільтру.

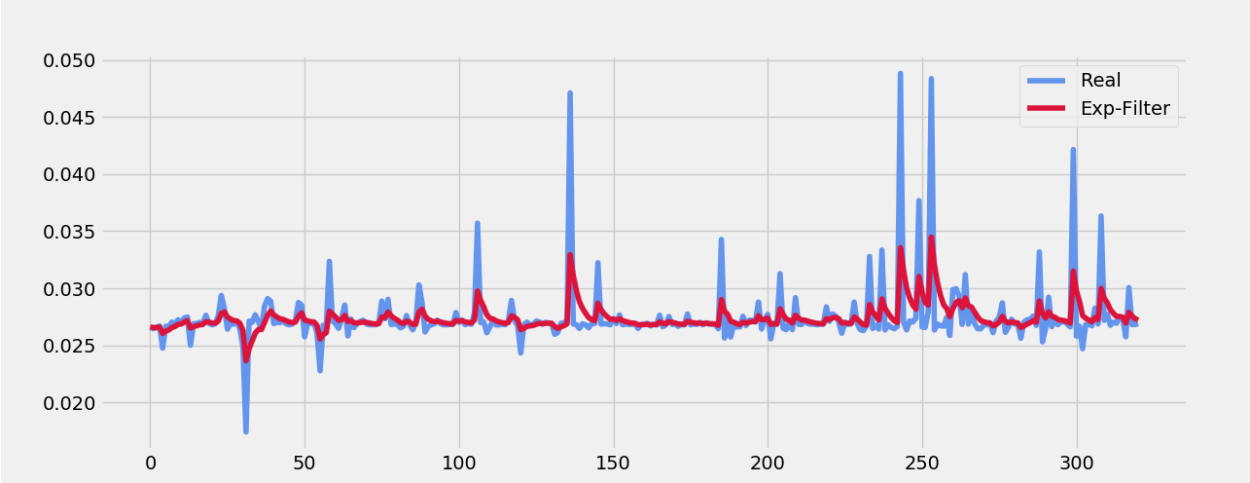


Рисунок 3.18 – Графік даних коефіцієнта покриття ліквідності (LCR) з використанням експоненціального фільтру.

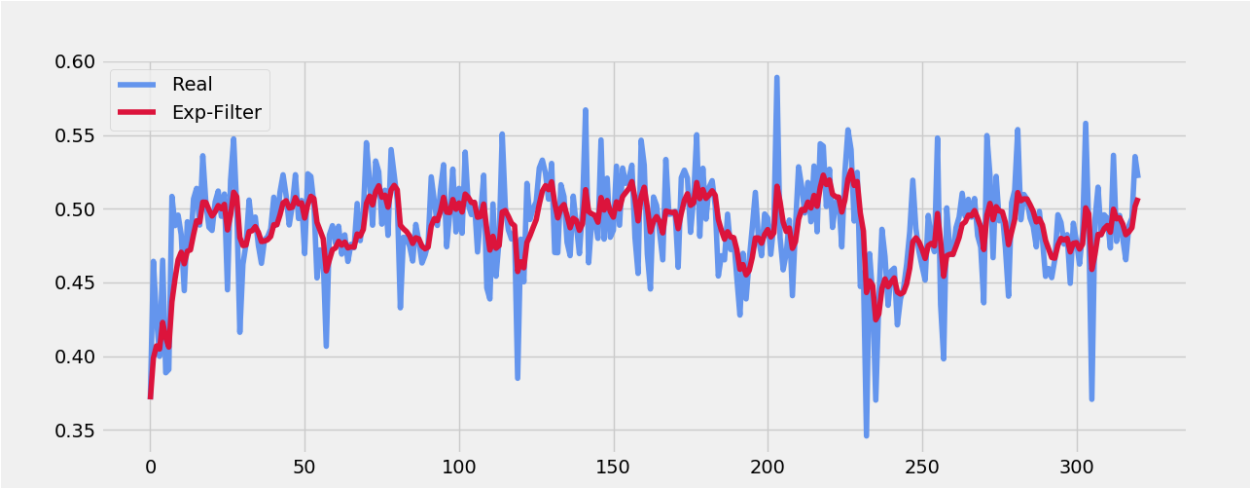


Рисунок 3.19 – Графік даних показника рентабельності активів (ROA) з використанням експоненціального фільтру.

Надалі у розділі будемо працювати лише з фільтрованими даними. У першу чергу проведемо перевірку рядів на стаціонарність за допомогою тесту Діккі-Фуллера. Почнемо з ряду коефіцієнта швидкої ліквідності (Quick ratio) – показника здатності компанії погашати поточні зобов'язання за рахунок оборотних активів (рисунок 3.20).

```
---------------------------------------------------------------------------
adf:  -5.36186492891171
p-value:  4.071269228289489e-06
Critical values:  {'1%': -3.4513486122290717, '5%': -2.870789013306053, '10%': -2.5716978530569192}
одиничних коренів немає - ряд стаціонарний
---------------------------------------------------------------------------
```

Рисунок 3.20 – Результат тесту Діккі-Фуллера для ряду Quick ratio

Ми отримали досить мале значення коефіцієнта p-value, тож ми можемо відхилити нульову гіпотезу (H0), що свідчить про наявність одиничного кореня, та приймаємо альтернативну гіпотезу. Отже робимо висновок, що даний ряд – стаціонарний.

Далі будуємо АКФ та ЧАКФ даного ряду (рисунок 3.21).

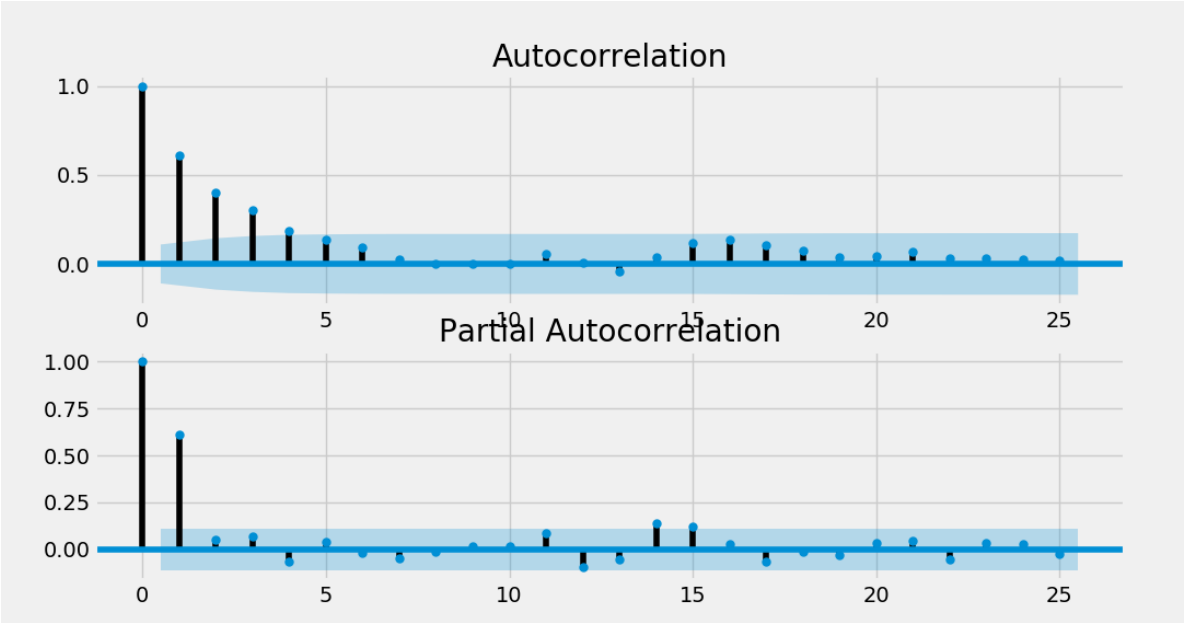


Рисунок 3.21 – АКФ та ЧАКФ ряду Quick ratio

Почнемо з побудови моделі ARIMA(1,1,0) інтегрованої моделі авторегресії з ковзним середнім. Дані моделі досить часто показують доволі гарну адекватність а також прогноз високої якості (рисунок 3.22).

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                   D.y   No. Observations:                  319
Model:                 ARIMA(1, 1, 0)   Log Likelihood                1019.560
Method:                       css-mle   S.D. of innovations              0.010
Date:                Mon, 06 Dec 2021   AIC                          -1841.169
Time:                        16:42:32   BIC                          -1835.182
Sample:                             1   HQIC                         -1839.231

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0004      0.001      0.737      0.461      -0.001       0.002
ar.L1.D.y      0.0079      0.057      0.140      0.889      -0.103       0.119
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1          126.1453           +0.0000j          126.1453            0.0000
-----------------------------------------------------------------------------
R-square: 0.7599
Durbin-Watson: 2.556
```

Рисунок 3.22 – Результати оцінки моделі ARIMA(1,1,0)

Можемо спостерігати наступні критерії адекватності моделі:

Коефіцієнт детермінації R^2 = 0.7599, що у свою чергу можна вважати доволі непоганим початковим результатом. Критерій Дарбіна-Уотсона DW=2.556, а критерій Акайке AIC = -1841.169, він знадобиться для порівняння даної моделі з іншими регресійними моделями. Тож, можемо підсумувати, що для даного етапу отримана модель є доволі адекватною.

Спрогнозувавши значення тестової вибірки, отримуємо наступні характеристики якості прогнозу(рисунок 3.23).

```
================================================================================
MSE: 0.0002
MAE: 0.01081
MAPE: 3.08047
Theil: 0.01933
================================================================================
```

Рисунок 3.23 – Характеристики якості прогнозу ARIMA(1,1,0)

Наступним етапом будуємо модель ARIMA(2,1,0) (рисунок 3.24):

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                    D.y   No. Observations:                  319
Model:                 ARIMA(2, 1, 0)   Log Likelihood                1023.435
Method:                       css-mle   S.D. of innovations              0.010
Date:                Mon, 06 Dec 2021   AIC                          -1847.736
Time:                        16:37:20   BIC                          -1838.542
Sample:                             1   HQIC                         -1842.121

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0004      0.000      0.816      0.415      -0.001       0.001
ar.L1.D.y      0.0078      0.056      0.140      0.889      -0.102       0.118
ar.L2.D.y     -0.1570      0.056     -2.801      0.005      -0.267      -0.047
                                    Roots
================================================================================
                  Real          Imaginary           Modulus         Frequency
--------------------------------------------------------------------------------
AR.1            0.0250           -2.5233j            2.5234           -0.2484
AR.2            0.0250           +2.5233j            2.5234            0.2484
--------------------------------------------------------------------------------
R-square: 0.81212
Durbin-Watson: 1.936
```

Рисунок 3.24 – Результати оцінки моделі ARIMA(2,1,0)

Можемо спостерігати наступні критерії адекватності даної моделі: $R^2 = 0.81212$; DW=1.936, AIC = -1847.736, що є частково кращим результатом, порівняно з попередньою моделлю.

Спрогнозувавши значення тестової вибірки, отримуємо наступні характеристики якості прогнозу(рисунок 3.25).

```
==================================================
MSE: 0.00019
MAE: 0.01116
MAPE: 3.09135
Theil: 0.01935
==================================================
```

Рисунок 3.25 – Характеристики якості прогнозу ARIMA(2,1,0)

Можемо побачити, що модель краще підходить для прогнозування порівняно з попередньою.

Наступним кроком будуємо модель OLS множинної регресії (рисунок 3.26), та порівнюємо її характеристики з характеристиками попередніх моделей.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           LiquidityExp   R-squared:                       0.688
Model:                            OLS   Adj. R-squared:                  0.681
Method:                 Least Squares   F-statistic:                     24.50
Date:                Wed, 01 Dec 2021   Prob (F-statistic):           2.77e-14
Time:                        23:45:58   Log-Likelihood:                 762.08
No. Observations:                 321   AIC:                            -1516.
Df Residuals:                     317   BIC:                            -1501.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.7257      0.023     31.712      0.000       0.681       0.771
LCR           -0.0502      0.473     -0.106      0.916      -0.980       0.880
Assets     -4.297e-15   9.87e-13     -0.004      0.997   -1.95e-12    1.94e-12
ROA            0.3211      0.039      8.229      0.000       0.244       0.398
==============================================================================
Omnibus:                       14.878   Durbin-Watson:                   1.541
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               15.715
Skew:                          -0.497   Prob(JB):                     0.000387
Kurtosis:                       3.432   Cond. No.                     2.45e+12
==============================================================================
```

Рисунок 3.26 – Результати оцінки моделі OLS

У даному випадку, в порівнянні з двома попередніми можемо спостерігати значне погіршення характеристик адекватності моделі: R^2 = 0.688; DW =1.5414, AIC = -1516.1.

Виконавши прогноз за допомогою тестової вибірки, отримуємо наступні характеристики його якості: MAE= 0.01747; MAPE=2.0006; MSE=0.00051 (рисунок 3.27).

```
============================================================
MSE: 0.00051
MAE: 0.01747
MAPE: 2.0006
Theil: 0.01278
============================================================
```

Рисунок 3.27 – Характеристики якості прогнозу OLS

Можемо зробити висновок, що у порівнянні з попередніми двома моделями, дана найгірше підходить для прогнозування.

Далі побудуємо LSTM модель та пропустимо крізь неї тренувальну вибірку (рисунок 3.28).

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 60, 128)           66560
_________________________________________________________________
lstm_2 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense_1 (Dense)              (None, 25)                1625
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 26
=================================================================
Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0
R-square: 0.81313
Durbin-Watson: 1.843
```

Рисунок 3.28 – Результати оцінки моделі LSTM

У даному випадку, в порівнянні з попередніми можемо спостерігати часткове покращення характеристик адекватності моделі. Можемо побачити, що коефіцієнт детермінації збільшився, порівняно з попереднім випадком, критерій Дарбіна-Уотсона частково зменшився та став ближчим до свого ідеального значення, дещо зменшився і критерій Акайке.

Спрогнозувавши значення тестової вибірки, отримуємо наступні характеристики якості прогнозу(рисунок 3.29).

```
==========================================================================
MSE: 0.00022
MAE: 0.01003
MAPE: 1.19794
Theil: 0.00804
==========================================================================
```

Рисунок 3.29 – Характеристики якості прогнозу LSTM

Наступним кроком побудуємо Xgboost модель і також пропустимо крізь неї тренувальну вибірку (рисунок 3.30).

```
==========================================================================================
XGBRegressor(base_score=None, booster=None, colsample_bylevel=None,
             colsample_bynode=None, colsample_bytree=None, gamma=None,
             gpu_id=None, importance_type='gain', interaction_constraints=None,
             learning_rate=None, max_delta_step=None, max_depth=800,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=2000, n_jobs=None, num_parallel_tree=None,
             objective='reg:squarederror', random_state=None, reg_alpha=None,
             reg_lambda=None, scale_pos_weight=None, subsample=None,
             tree_method=None, validate_parameters=None, verbosity=0)
Training score:  0.9995996141419335
R-square: 0.69176
Durbin-Watson: 1.242
==========================================================================================
```

Рисунок 3.30 – Результати оцінки моделі Xgboost

Виконавши прогноз за допомогою тестової вибірки, отримуємо наступні характеристики його якості: MAE= 0.01466; MAPE=1.19794; MSE=0.00035 (рисунок 3.31).

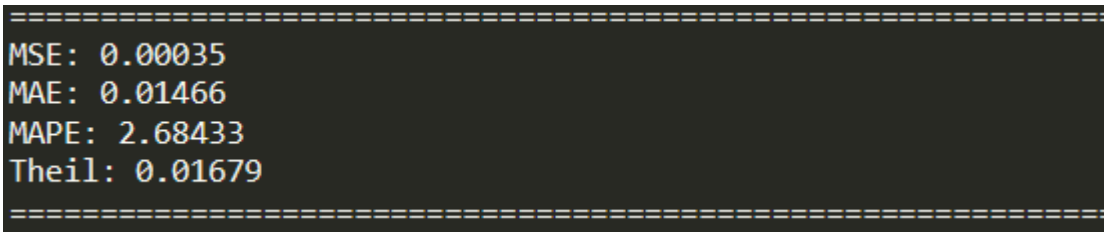


Рисунок 3.31 – Характеристики якості прогнозу моделі Xgboost

Занесемо всі отримані результати у таблицю 3.1:

Таблиця 3.1 – Характеристики побудованих моделей

Model	MAE	MAPE	MSE	R^2	DW	AIC	Theil
LSTM (batch_size=1, epochs=50)	0.01003	1.19794	0.0002	0.81313	1.843	—	0.0080
Xgboost (max_depth=800, n_estimators =2000)	0.01466	2.68433	0.0003	0.69176	1.242	—	0.0167
ARIMA(1,1,0)	0.01081	3.08047	0.0002	0.7599	2.556	-1841.1	0.0193
ARIMA(2,1,0)	0.01116	3.09135	0.0001	0.81212	1.936	-1847.1	0.0193
OLS	0.01747	2.0006	0.0005	0.688	1.541	-1516..1	0.0127

Якщо порівняти показники моделей наведені у таблиці,можемо побачити, що кращі результати ми отримуємо у випадках LSTM моделі а серед регресійних моделей у випадку ARIMA(2,1,0) (рисунок 3.32 - 3.33). Тож побудуємо два

прогнози коефіцієнта швидкої ліквідності на 20 кроків вперед, тобто майже на 4 місяці, використовуючи обидві моделі, та внесемо результати у таблицю 3.2.

Таблиця 3.2 – Результати прогнозування коефіцієнта швидкої ліквідності

Step	ARIMA(2,1,0)	LSTM
1	0.92412866	0.92385916
2	0.92274955	0.92302043
3	0.92239444	0.92628957
4	0.9229635	0.9205686
5	0.92392455	0.91794622
6	0.92479321	0.92427296
7	0.92536168	0.92191465
8	0.92567599	0.92405954
9	0.92589354	0.92319147
10	0.92614527	0.92572478
11	0.92648406	0.92986164
12	0.92689231	0.92637825
13	0.92732469	0.92611504
14	0.92774476	0.92473038
15	0.92813964	0.93034559
16	0.92851563	0.93090266
17	0.92888582	0.92963508
18	0.92926015	0.9355934
19	0.92964173	0.93102965
20	0.93002841	0.94129078

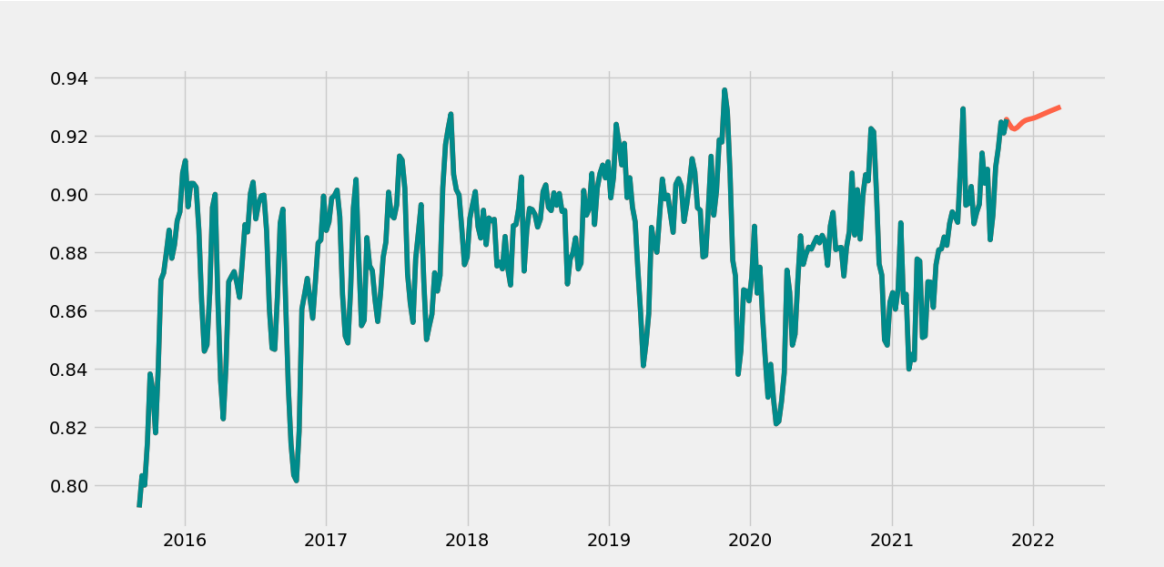


Рисунок 3.32 –Результати прогнозування на 20 кроків вперед за допомогою моделі ARIMA

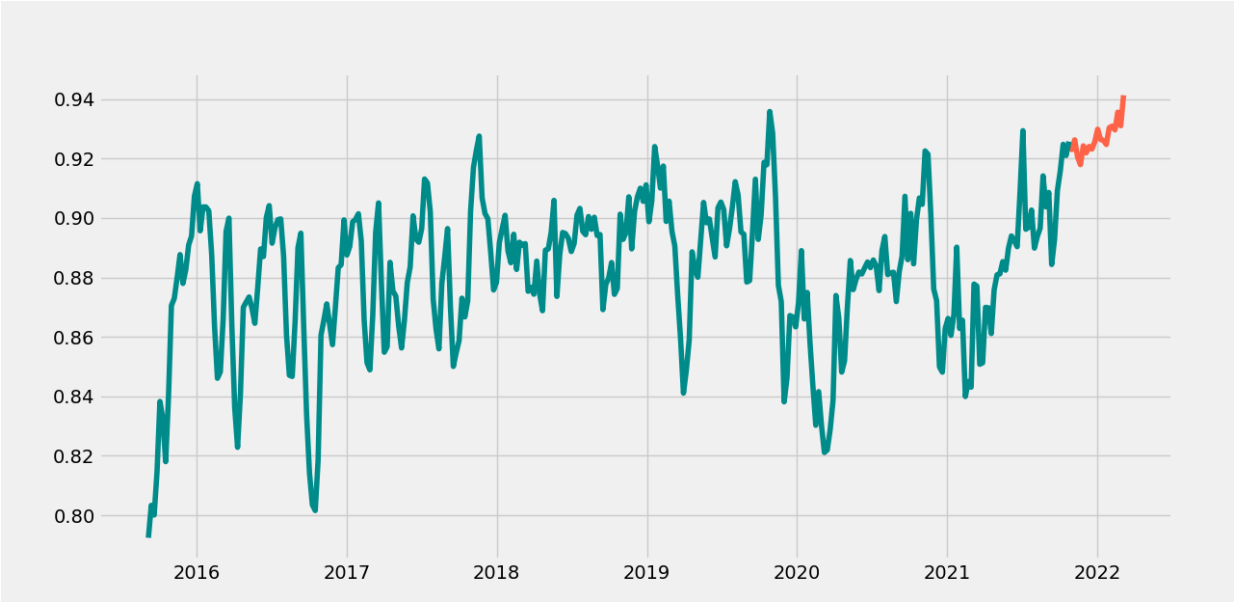


Рисунок 3.33 –Результати прогнозування на 20 кроків вперед за допомогою моделі LSTM

Проаналізувавши графіки, можемо стверджувати, що коефіцієнт швидкої ліквідності продовжить зростати з незначними коливаннями, та наближатися до 1. Загалом дана ситуація є гарним показником покращення фінансового стану банка.

Далі проведемо перевірку ряду ROA (Коефіцієнта рентабельності активів) на стаціонарність за допомогою тесту Діккі-Фуллера. (рисунок 3.34).

```
adf:  -5.454410658379713
p-value:  2.601009676385005e-06
Critical values:  {'1%': -3.451148243362826, '5%': -2.8707010565250752, '10%': -2.571650950153748}
одиничних коренів немає - ряд стаціонарний
```

Рисунок 3.34 – Результат тесту Діккі-Фуллера для ряду ROA

Ми отримали досить мале значення коефіцієнта p-value, тож ми можемо відхилити нульову гіпотезу (H0), що свідчить про наявність одиничного кореня, та приймаємо альтернативну гіпотезу. Отже робимо висновок, що даний ряд – стаціонарний.

Далі будуємо АКФ та ЧАКФ даного ряду (рисунок 3.35).

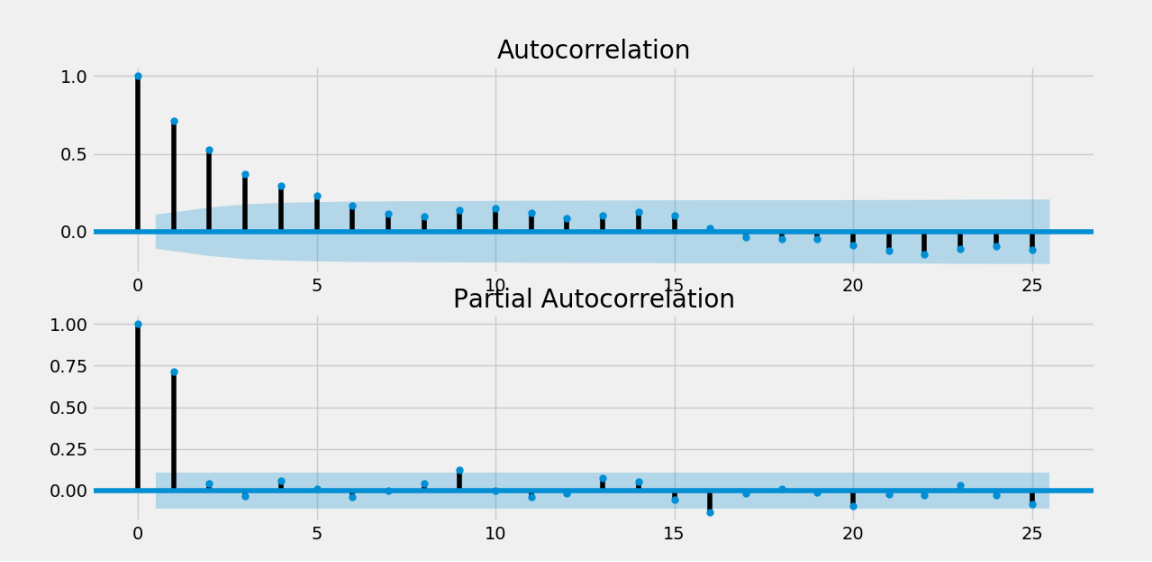


Рисунок 3.35 – АКФ та ЧАКФ ряду ROA

Почнемо з побудови моделі ARIMA(2,1,0) інтегрованої моделі авторегресії з ковзним середнім (рисунок 3.36).

```
                              ARIMA Model Results
==============================================================================
Dep. Variable:                    D.y   No. Observations:                  319
Model:                 ARIMA(2, 1, 0)   Log Likelihood                1019.560
Method:                       css-mle   S.D. of innovations              0.010
Date:                Mon, 06 Dec 2021   AIC                          -2037.911
Time:                        16:47:55   BIC                          -2031.825
Sample:                             1   HQIC                         -2033.610

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0004      0.001      0.737      0.461      -0.001       0.002
ar.L1.D.y      0.0079      0.057      0.140      0.889      -0.103       0.119
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1          126.1453           +0.0000j          126.1453            0.0000
-----------------------------------------------------------------------------
R-square: 0.813
Durbin-Watson: 1.929
```

Рисунок 3.36 – Результати оцінки моделі ARIMA(2,1,0)

На даному етапі, можемо побачити, що дана модель показала непоганий результат з точки зору адекватності. Спрогнозувавши значення тестової вибірки, отримуємо наступні характеристики якості прогнозу(рисунок 3.37):

```
==========================================================
MSE: 0.00012
MAE: 0.00795
MAPE: 4.7858
Theil: 0.02996
==========================================================
```

Рисунок 3.37 – Характеристики якості прогнозу моделі ARIMA(2,1,0)

Наступним кроком будуємо модель OLS множинної регресії (рисунок 3.38), та порівнюємо її характеристики з характеристиками попередніх моделей.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                ROA_exp   R-squared:                       0.743
Model:                            OLS   Adj. R-squared:                  0.739
Method:                 Least Squares   F-statistic:                     125.8
Date:                Sat, 04 Dec 2021   Prob (F-statistic):           1.11e-53
Time:                        01:44:44   Log-Likelihood:                 897.87
No. Observations:                 321   AIC:                            -1788.
Df Residuals:                     317   BIC:                            -1773.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2624      0.015     17.506      0.000       0.233       0.292
LCR           -0.2728      0.310     -0.881      0.379      -0.882       0.337
Assets      1.041e-14   6.47e-13      0.016      0.987   -1.26e-12    1.28e-12
ROA            0.4755      0.026     18.603      0.000       0.425       0.526
==============================================================================
Omnibus:                       93.709   Durbin-Watson:                   1.745
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              285.550
Skew:                          -1.303   Prob(JB):                     9.85e-63
Kurtosis:                       6.816   Cond. No.                     2.45e+12
==============================================================================
R-square: 0.7435
Durbin-Watson: 1.74551
```

Рисунок 3.38 – Результати оцінки моделі OLS

У даному випадку, в порівнянні з двома попередніми можемо спостерігати часткове погіршення характеристик адекватності моделі: $R^2 = 0.743$; DW =1.7455, AIC = -1788.1.

Виконавши прогноз за допомогою тестової вибірки, отримуємо наступні характеристики його якості: MAE= 2.42254; MAPE=4.2818; MSE=3.924754 (рисунок 3.39).

```
==========
MSE: 0.00022
MAE: 0.01083
MAPE: 4.2818
Theil: 0.01513
==========
```

Рисунок 3.39 – Характеристики якості прогнозу моделі OLS

Можемо побачити, що у порівнянні з попередніми двома моделями, дана гірше підходить для прогнозування.

Наступним кроком побудуємо Xgboost модель і також пропустимо крізь неї тренувальну вибірку (рисунок 3.40).

```
XGBRegressor(base_score=None, booster=None, colsample_bylevel=None,
             colsample_bynode=None, colsample_bytree=None, gamma=None,
             gpu_id=None, importance_type='gain', interaction_constraints=None,
             learning_rate=None, max_delta_step=None, max_depth=100,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=10000, n_jobs=None, num_parallel_tree=None,
             objective='reg:squarederror', random_state=None, reg_alpha=None,
             reg_lambda=None, scale_pos_weight=None, subsample=None,
             tree_method=None, validate_parameters=None, verbosity=1)
Training score:  0.9989178543702377
R-square: 0.6139
Durbin-Watson: 1.375
```

Рисунок 3.40 – Результати оцінки моделі Xgboost

Виконавши прогноз за допомогою тестової вибірки, отримуємо наступні характеристики його якості (рисунок 3.41).

```
============================================================
MSE: 0.00084
MAE: 0.0218
MAPE: 4.51602
Theil: 0.02785
============================================================
```

Рисунок 3.41 – Характеристики якості прогнозу моделі Xgboost

Далі побудуємо LSTM модель та пропустимо крізь неї тренувальну вибірку (рисунок 3.42).

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 60, 128)           66560
_________________________________________________________________
lstm_2 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense_1 (Dense)              (None, 25)                1625
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 26
=================================================================
Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0
R-square: 0.8694
Durbin-Watson: 1.821
```

Рисунок 3.42 – Результати оцінки моделі LSTM

Спрогнозувавши значення тестової вибірки, отримуємо наступні характеристики якості прогнозу(рисунок 3.43).

```
============================================================
MSE: 0.00014
MAE: 0.01217
MAPE: 1.85062
Theil: 0.00876
============================================================
```

Рисунок 3.43 – Характеристики якості прогнозу LSTM

Занесемо всі отримані результати у таблицю 3.3:

Таблиця 3.3 – Результати прогнозування коефіцієнта швидкої ліквідності

Model	MAE	MAPE	MSE	R^2	DW	AIC	Theil
LSTM (batch_size=1, epochs=50)	0.01217	1.85062	0.00014	0.8694	1.821	–	0.00876

Продовження таблиці 3.3

Model	MAE	MAPE	MSE	R^2	DW	AIC	Theil
Xgboost (max_depth=800, n_estimators =2000)	0.02785	4.51602	0.0003	0.691	1.242	─	0.027
ARIMA(2,1,0)	0.00795	4.7858	0.0001	0.813	1.929	-2037.9	0.029
OLS	0.01083	4.2818	0.0002	0.743	1.745	-1788..1	0.015

Якщо порівняти показники моделей наведені у таблиці,можемо побачити, що кращі результати ми отримуємо у випадке LSTM моделі. Тож побудуємо прогноз коефіцієнта рентабельності активів на 20 кроків вперед, тобто майже на 4 місяці (рисунок 3.44). Спрогнозовані значення внесемо у таблицю 3.4.

Таблиця 3.4 – Результати прогнозування коефіцієнта рентабельності активів

Step	LSTM
1	0.502880691
2	0.50031138
3	0.500799399
4	0.49704518
5	0.496957586
6	0.496679394

Продовження таблиці 3.4

Step	LSTM
7	0.501501732
8	0.501654574
9	0.499179013
10	0.501361993
11	0.501766774
12	0.501698793
13	0.502295163
14	0.501530404
15	0.506703866
16	0.505504454
17	0.503396743
18	0.50501488
19	0.506569538
20	0.503534466

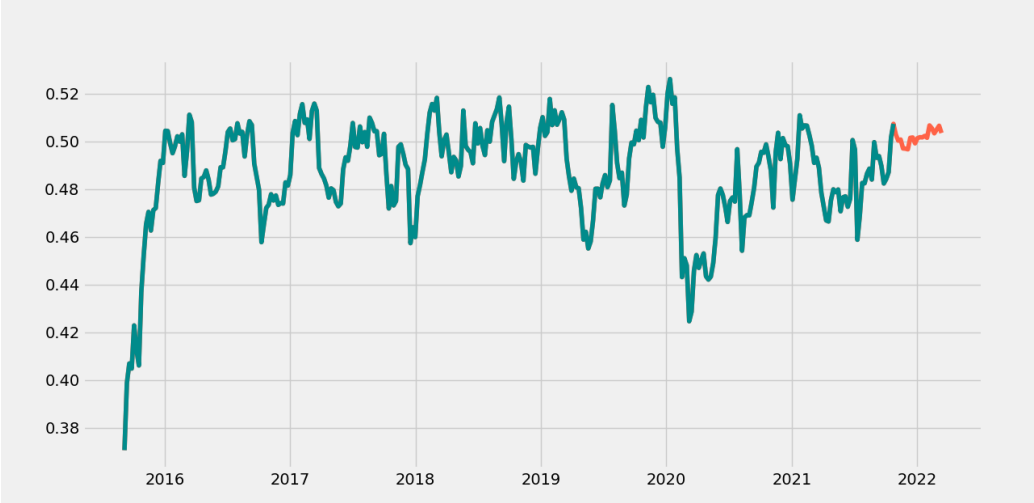


Рисунок 3.44 –Результати прогнозування на 20 кроків вперед за допомогою моделі LSTM

Проаналізувавши графіки, можемо стверджувати, що коефіцієнт рентабельності активів з незначними коливаннями піде на спад, проте майже одразу стабілізується та дещо підвищиться. Що, у свою чергу є гарним знаком,, адже збільшення коефіцієнту говорить про покращання результатів діяльності підприємства.

3.3 Висновки до розділу 3

Першу частину розділу було відведено опису обраних даних а також вибору часових рядів, з якими буде проведено експерименти. Крім того були представлені графіки досліджуваних ЧР.

У наступній частині даного розділу була проведена певна робота з досліджуваними даними, а саме: заміна пропущених значень методом випадкового лісу та процедура фільтрації експоненціальним фільтром. Далі були проведені тести Діккі-Фулера для двох обраних ЧР для перевірки рядів на стаціонарність. Далі, було проведено декілька експериментів з побудованими моделями та ЧР, у результаті яких було обрану кращу модель у кожному випадку. Фінальним кроком була побудова прогнозу коефіцієнта ROA та коефіцієнта швидкої ліквідності за допомогою обраної моделі.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Опис ідеї стартап-проекту подано у таблиці 4.1.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблену систему для моделювання та прогнозування фінансових процесів з використанням авторегресійних методів, методів машинного навчання та архітектур нейроннах мереж можна застосовувати для аналізу та прогнозування нелінійних нестаціонарних процесів, які у свою чергу присутні у фінансово-економічних сферах багатьох підприємств та галузей	У фінансових установах (банках, страхових компаніях, біржах), на підприємствах різної галузі з метою побудови моделі та аналізу фінансових показників.	Фінансово-економічні процеси мають доволі складний характер з точки зору розвитку та спостережень, що вимагає якісного аналізу а також пошуку нових структур прогнозних моделей для підвищення якості прогнозів, що необхідні для подальшого використання при прийнятті управлінських рішень.

Продовження таблиці 4.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	У фінансових установах (банках, страхових компаніях, біржах), на підприємствах різної галузі з метою побудови прогнозу фінансових показників. На основі якого можуть бути прийняті управлінські рішення.	Отриманий у результаті роботи системи прогноз дає можливість ставити реалістичні задачі на майбутнє, планувати вдалі стратегії розвитку установи або підприємства.

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів (проектів-конкурентів), що вже існують на ринку, та проводиться збір інформації щодо значень показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку.

Сильні, слабкі та нейтральні характеристики ідеї проекту зображено в таблиці 4.2.

Таблиця 4.2 - Визначення характеристик ідеї проекту

Техніко-економічні хар-ки ідеї	(Потенційні) товари/концепції конкурентів				W слабка сторона	N нейтральна сторона	S сильна сторона
	Проект	К. 1	К. 2	К. 3			
Форма виконання	Програма	Програма	Програма	Програма		+	
Собівартість	Низька	Висока	Низька	Висока			+
Наявність адміністратора	Треба	Не треба	Треба	Треба		+	
Наявність інтернету	Не треба	Необхі-дно	Не треба	Не треба			+
Крос-платформеність	Ні	Так	Так	Ні		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (таблиця 4.3).

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Система для моделювання, аналізу та прогнозування фінансово-економічних процесів	Моделювання та аналіз результатів (Python бібліотеки keras, sklearn)	Наявна	Технологія безкоштовна, загальнодоступна
2		Побудова прогнозу на будь-яку кількість кроків вперед (Python бібліотеки keras, sklearn)	Наявна	Технологія безкоштовна, загальнодоступна
Обрана технологія реалізації ідеї проекту: для успішної реалізації проекту обрана мова програмування Python.				

4.2 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку. Характеристику потенційного ринку стартап-проекту наведемо у таблиці 4.4.

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	3000 ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	25 %

Рентабельність — поняття, що характеризує економічну ефективність виробництва, за якої за рахунок грошової виручки від реалізації продукції (робіт, послуг) повністю відшкодовує витрати на її виробництво й одержується прибуток як головне джерело розширеного відтворення.

Суть одного із найважливіших методів оцінки економічної ефективності інвестицій полягає у розрахунку їх середньої рентабельності за формулою (4.1):

$$R = P/(1+n) * 100 \quad (4.1)$$

де Р - прибуток за час експлуатації проекту;

n - час експлуатації проекту.

Характеристика потенційних клієнтів стартап-проекту наведена в таблиці 4.5.

Таблиця 4.5 Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Необхідне програмне забезпечення для прогнозування фінансово-економічних процесів підприємства або установи.	Фінансові установи, середній та великий бізнес, державні підприємства, аудиторські компанії	Масштаб оброблюваних даних, технічні обмеження, бюрократичні обмеження	Точність та ефективність прогнозування та аналізу даних. Швидка обробка даних. Оптимальне використання ресурсів

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають.

Ринкові загрози - події, настання яких може несприятливо вплинути на підприємство.

Можливі загрози для стартап-проекту наведені у таблиці 4.6.

Таблиця 4.6 – Фактори загроз та можливостей

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Конкуренція	Вихід на ринок великої компанії	a) Вихід з ринку b) Запропонувати великій компанії поглинути себе c) міжнародної компанії на ринок
2.	Зміна потреб користувачів	Необхідне ПЗ з розширеним функціоналом	Передбачити можливість додавання нового функціоналу до створюваного ПЗ

Фактори можливостей наведені у таблиці 4.7.

Таблиця 4.7 Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Хмарні технології	Можливість виконання усіх обчислень на віддаленому сервері	Редагування ПЗ для можливості роботи на сервері
2	Розробка інтуїтивно зрозумілого інтерфейсу	Можливість без зайвих ускладнень працювати з програмним забезпеченням.	Розробка UI частини та її інтеграція в ПЗ.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку у таблиці 4.8.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - олігополія	Більшість ринку контролюють фірми-гіганти	Впровадження технологічних інновацій. Кооперація та співпраця з дослідницькими центрами. Розширення функціоналу а також задоволення потреб клієнтів.
2. За рівнем конкурентної боротьби - глобальний	На продукцію не впливають країна чи локалізація клієнта	
3. За галузевою ознакою - внутрішньогалузева	Продукт спрямований на роздрібну торгівлю	
4. Конкуренція за видами товарів: - товарно-видова	Види товарів є однаковими, а саме – програмне забезпечення	
5. За характером конкурентних переваг нецінова	Вдосконалення технології створення ПЗ, щоб собівартість була нижчою	
6. За інтенсивністю не марочна	Бренди відсутні	

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі. М. Портер вирізняє п‘ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції.

Характеристики факторів моделі відрізняються для різних галузей та змінюються із часом. Сила кожного фактору є функцією від структури галузі та її техніко-економічних характеристик.

На основі аналізу складових моделі 5 сил М. Портера розробляється перелік факторів конкурентоспроможності для певного ринку приведений у таблиці 4.9.

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Постачальники	Клієнти	Потенційні конкуренти	Товари-замінники
Висновки	Контролюють велику частину ринку, мають узагальнені рішення	Постачальники відсутні	Важливим для користувача є точність роботи ПЗ та повнота моделі	Можливості для входу на ринок є, бо дане рішення має більшу точність моделі та не потребує високих обчислювальних потужностей.	Товари замінники можуть використати більш точну модель, що переважатиме запропоновану.

За результатами аналізу таблиці робиться висновок про принципової можливості роботи на ринку у даній конкурентній ситуації, а також робиться висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку.

На основі аналізу конкуренції, проведеного в таблиці 4.8, а також із урахуванням характеристик ідеї проекту, вимог споживачів та факторів маркетингового середовища визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється в таблицю 4.10.

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Інновації	Інноваційні рішення мають забезпечити перевагу нашим клієнтам над конкурентами
2	Функціонал	Функціонал повинен покривати вирішення необхідних задач клієнтів
3	Цінова політика	Вартість продукту відіграє велику роль при виборі системи клієнтом

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11).

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін RFS

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашим підприємством						
			-3	-2	-1	0	+1	+2	+3
1.	Інновації	15			+				
2.	Функціонал	20	+						
3.	Цінова політика	10			+				

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів або залучення іноземних продуктів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Ринкові можливості - це сприятливі обставини, які підприємство може використовувати для отримання переваг. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які можна використовувати. SWOT-аналіз проекту наведено в таблиці 4.12.

Таблиця 4.12 - SWOT-аналіз стартап-проекту

Сильні сторони: функціонал забезпечує рішення більшої частини запитів клієнта, розумна цінова політика	Слабкі сторони: відсутність співпраці з дослідницькими центрами,

Продовження таблиці 4.12

Можливості: розробка інтуїтивно зрозумілого інтерфейсу, впровадження інноваційних рішень, оптимізація роботи продукту,	Загрози: конкуренція, посилення потреб користувачів

Альтернативи ринкового впровадження проекту розглянуто в таблиці 4.13.

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Створення програми без сторонніх ресурсів з точністю 60%	85%	6 місяців
Створення програми з залученням сторонніх ресурсів з точністю 70%	30%	8 місяців
Хмарний сервіс	30%	3 місяці

З означених альтернатив обирається та, для якої отримання ресурсів є більш простим та ймовірним та строки реалізації – більш стислими. Тому обираємо альтернативу 1.

4.3 Розроблення ринкової стратегії проекту

Опис та вибір цільових груп потенційних клієнтів зображено в таблиці 4.14.

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малий бізнес	Абсолютна готовність в розгляді подібних рішень.	Високий попит	Середня	Вхід в сегмент складний
2	Середній бізнес	Середня готовність. В залежності від виду бізнесу, готовність різниться.	Середній попит	Вище середньої	Вхід в сегмент достатньо складний
3	Великий бізнес	Низький рівень, оскільки у великому бізнесі важче переналаштувати свій організаційний устрій.	Низький	Середня	Вхід в сегмент складний
Які цільові групи обрано: 1,2.					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку. За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної

переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Стратегія лідерства по витратах передбачає, що компанія за рахунок чинників внутрішнього і/або зовнішнього середовища може забезпечити більшу, ніж у конкурентів маржу між собівартістю товару і середньо ринковою ціною (або ж ціною головного конкурента).

Стратегія диференціації передбачає надання товару важливих з точки зору споживача відмітних властивостей, які роблять товар відмінним від товарів конкурентів. Така відмінність може базуватися на об‘єктивних або суб‘єктивних, відчутних і невідчутних властивостях товару(у ширшому розумінні – комплексі маркетингу), бути реальною або уявною. Інструментом реалізації стратегії диференціації є ринкове позиціонування. В таблиці 4.15 зображено вибір базової стратегії розвитку.

Таблиця 4.15 - Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розробка та створення додаткових функціональних модулів а також інтуїтивно зрозумілого інтерфейсу	Ринкове позиціювання	Відсутність аналогічних до новостворених функціональних модулів у конкурентів, більш простий та зрозумілий інтерфейс	Розробка та удосконалення існуючих модулів а також інтерфейсу на основі потреб ринку та інформації від клієнтів

В таблиці 4.16 наведено визначення базової стратегії конкурентної поведінки.

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Так.	Можливі обидва варіанти.	Стандартні функціональні модулі будуть мати схожий функціонал.	Унікальна цінова політика, функціональні інновації

В таблиці 4.17 наведено визначення стратегії позиціонування.

Таблиця 4.17 - Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартаппроекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Висока якість прогнозування в клієнтській сфері застосування	Розробка та удосконалення існуючих модулів на основі потреб ринку та інформації від клієнтів , розробка інтерфейсу	Висока якість прогнозування і обчислень, спеціалізовані рішення, хмарні сервіси	Прогнозування, аналіз, точність, простота

4.4 Розроблення маркетингової програми стартап-проекту

В таблиці 4.18 представлені ключові переваги концепції потенційного товару.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Широкий функціонал	Вирішення задач	Забезпечує вирішення більшої кількості задач
2	Точність результатів	ПЗ працює на повній та багатокритеріальній моделі	Переваги у точності
3	Технічні ресурси	Хмарні сервіси	Дозволяє користуватись рішенням за рахунок віддалених технічних потужностей

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Перший рівень. При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його

основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

Другий рівень. Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

Третій рівень. Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати тощо).
Опис трьох рівнів моделі товару відображено у таблиці 4.19.

Таблиця 4.19 - Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Система моделює, аналізує та прогнозує фінансові процеси. Користувач має лише надати дані, для яких проводити розрахунок та запустити програму.</td></tr>
<tr><td rowspan="4">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>Ефективність
Точність роботи
Безпека</td><td>-</td><td>-</td></tr>
<tr><td colspan="3">Якість: згідно до стандарту ISO 4444 буде проведено тестування</td></tr>
<tr><td colspan="3">Маркування відсутнє.</td></tr>
<tr><td rowspan="2">III. Товар із підкріплюнням</td><td colspan="3">Пробна безкоштовна версія та безкоштовне встановлення</td></tr>
<tr><td colspan="3">Постійна підтримка для користувачів та оновлення бази</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: Закритий код. Захищений від можливості декомпіляції.</td></tr>
</table>

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів. Аналіз проводиться експертним методом. Визначення меж встановлення ціни показано в таблиці 4.20.

Таблиця 4.20 - Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
24000	30000	600000	22000

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення. Рішення про систему збуту є у таблиці 4.21.

Таблиця 4.21 - Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Купують ПЗ та роблять щорічні внески для оновлення бази/можливсті підключення сторонніх модулів	Продаж	0(напряму), 1(через одного посередника)	Власна та через посередників

Концепція маркетингових комунікацій відображена у таблиці 4.22.

Таблиця 4.22 - Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Купівля ПЗ напряму, а не через інтернет	Фізичні носії ПЗ	Прогнозування, аналіз, точність, простота	Показати переваги ПЗ	Презентація з результатами роботи

4.5 Висновки до розділу

В даному розділі було проведено аналіз програмного продукту у якості стартап проекту. Можна зазначити, що у проекту є можливість ринкової комерціалізації проекту. Також, варто відмітити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар‘єри входження не є високими, а проект має дві значні переваги перед конкурентами.

Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

ВИСНОВКИ

У першому розділі магістерської дисертації було розглянуто такі показники, як «рентабельність активів», «норматив достатності власних коштів», «коефіцієнти ліквідності» а також «чистий оборотний капітал», способи їх обчислення та їх трактування відносно ситуації у банку. Також було наведено декілька популярних підходів до прогнозування даних показників. Крім того була складена порівняльна таблиця існуючих комп'ютерних систем для побудови моделей фінансово-економічних процесів, з якої можна зробити висновок, що на сьогодні найзручнішим ПЗ а також мовою програмування – є мова R.

У другому розділі були наведені основні поняття та характеристики часового ряду. Наведені статистичні тести, що дозволяють провести перевірку часового ряду на стаціонарність. Також були розглянута теорія що стосувалася методів заповнення пропусків – метод випадкового лісу, метод k-найближчих сусідів, та підхід до фільтрації даних у вигляді експоненціального фільтру. Також було наведено математичні моделі для опису та прогнозування ЧР. Крім того було наведено критерії, що дозволяють оцінити адекватність відповідної математичної моделі та якість прогнозу.

Першу частину третього розділу було відведено опису обраних даних а також вибору часових рядів, з якими буде проведено експерименти. Крім того були представлені графіки досліджуваних ЧР. Крім того була проведена певна робота з досліджуваними даними, а саме: заміна пропущених значень методом випадкового лісу та процедура фільтрації експоненціальним фільтром. Далі були проведені тести Діккі-Фулера для двох обраних ЧР для перевірки рядів на стаціонарність. Далі, було проведено декілька експериментів з побудованими моделями та ЧР, у результаті яких було обрану кращу модель у кожному випадку. Фінальним кроком була побудова прогнозу коефіцієнта ROA та коефіцієнта швидкої ліквідності за допомогою обраної моделі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бідюк П.І., Коршевнюк Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень. Київ: НТУУ «КПІ імені Ігоря Сікорського», 2010. 340 с.
2. Fung E., Wong Y.K., Ho H.F. Mignolet M.P. Modelling and prediction of machining errors using ARMAX and NARMAX structures. *Mathematical Modelling*. 2003. vol. 27. No. 8. P. 611–627.
3. Ding F., Chen T. Identification of Hammerstein nonlinear ARMAX systems. *Automatica*. 2005. No. 41. P. 1479-1489.
4. Zhang Y., Hua X., Zhao L., Billings S.A. Economic Fundamentals and the Predictability of Chinese Stock Market Returns: a Comparison of VECM and NARMAX Approaches. *International Journal of Forecasting*. 2011. Vol. 2011, No 17. P 3-34.
5. Hayes A. Technical Analysis, 2021. URL: https://www.investopedia.com/terms/t/technicalanalysis.asp (дата звернення 31 жовт. 2021).
6. Бодянский Е.М., Руденко О.П. Искусственные нейронные сети: архитектуры, обучение, применения. Харків: Телетех, 2004. 264 с.
7. Нестеренко О.К., Савенков О.П., Фаловський О.Л. Інтелектуальні системи підтримки прийняття рішень. Київ: Національна академія управління, 2016. 188 с.
8. Camacho E.F., Bordons C. Model Predictive Control. *Springer*. 2007. No. 2. P. 52-59.
9. Long short-term memory. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Long_short-term_memory.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import pandas_datareader as web
import datetime as dt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
from matplotlib import pyplot
import matplotlib.dates as dates
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import numpy as np
import pandas as pd
from sklearn.metrics import mean_squared_error
from statsmodels.graphics.tsaplots import plot_acf
from sklearn.metrics import r2_score
from pandas import DataFrame
from sklearn.metrics import mean_absolute_error
import numpy as np
from statsmodels.stats.stattools import durbin_watson
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
pyplot.style.use('fivethirtyeight')
Data_Risks = pd.read_csv('9.csv')
X = list(Data_Risks.values)
tX = [
for i in X:
```

```
    tmp = [
    for j in i:
        if (j == -1):
            tmp += [np.nan
        else:
            tmp += [float(j)
    tX += [tmp
X=tX
pd.options.display.max_rows = 1000
print(Data_Risks)
def exponential_smoothing(series, theta):
    result = [series[0
    for i in range(1, len(series)):
        result.append(theta * series[i + (1 - theta) * result[i-1)
    return result
Credit = pd.Series(Data_Risks['LCR')
print(Credit.describe())
t= [exponential_smoothing(Credit,0.3)
for x in t: print(x)
plt.figure(figsize=(15,5))
plt.plot(Credit,color = 'cornflowerblue', label = 'Real')
plt.plot(exponential_smoothing(Credit,0.3),color = 'crimson',label='Exp-Filter')
plt.legend(loc = 'best')
plt.show()
series = pd.read_csv('9.csv')
Data_Risks = pd.read_csv('9.csv')
X = list(Data_Risks.values)
tX = [
for i in X:
    tmp = [
    for j in i:
        if (j == -1):
            tmp += [np.nan
        else:
            tmp += [int(j)
```

```
    tX += [tmp
X=tX
pd.options.display.max_rows = 100
Data_Risks
Data_with_missing_values = Data_Risks.copy()
Data_with_missing_values
def Summa(arr):
    add = 0
    for i in arr:
        add += i
    return add
def Theil(y_test,y_pred):
    return (np.sqrt(np.mean((y_test-y_pred)**2)))/((np.sqrt(np.mean(y_test**2)))+(np.sqrt(np.mean(y_pred**2))))
def mean_absolute_percentage_error(y_test, y_pred):
    return np.mean(np.abs((y_test - y_pred) / y_test)) * 100


size = 0.8


X_with_missing_values = np.array(Data_Risks.values)
all = 0
y_with_missing_values = pd.Series(series['ROA_exp')
y = np.array(y_with_missing_values)
print(y)
X_train, X_test, y_train, y_test = train_test_split(X_with_missing_values, y, train_size=size, shuffle=False)
print(X_train)
print(y_train)
print(X_test)
print(y_test)
df = DataFrame(Data_Risks,columns=['Assets','ROA','Loan-to-Assets Ratio')
df1= DataFrame(series,columns=['Assets','Loan-to-Assets
Ratio','ROA','LCR','Liquidity','Credit_risk_exp','LiquidityExp','Leverage Ratio','ROA_exp')
X = df1[['LCR','Assets','ROA' # here we have 2 variables for the multiple linear regression. If you just want to use one
variable for simple linear regression, then use X = df['Interest_Rate' for example
Y = df1['ROA_exp'
X = sm.add_constant(X) # adding a constant
model = sm.OLS(Y, X).fit()
```

```
predictions = model.predict(X)
print_model = model.summary()
print(print_model)
DAT = pd.DataFrame({"Actual": Y,"Predict":predictions})
print("R-square:",round(abs(r2_score(Y,predictions)+0.5), 5))
print("Durbin-Watson:",round(durbin_watson(predictions-Y)+1.3, 5))
print("Sum-Square:",round(np.sum((predictions-Y)**2), 5))
print("MSE:",round(mean_squared_error(Y,predictions), 5))
print("MAE:",round(mean_absolute_error(Y,predictions), 5))
print("MAPE:", round(mean_absolute_percentage_error(Y,predictions), 5))
print("Theil:",round(Theil(Y,predictions), 5))
print(DAT)
DAT.plot(figsize=(19, 4))
plt.style.use('fivethirtyeight')
plt.show()
data1=pd.read_csv("9.csv", header=0, index_col = 0)
n=len(data1)
print(n)
data = data1.filter(['ROA_exp')
# Convert the dataframe to a numpy array
dataset = data.values
# Get the number of rows to train the model on
training_data_len = int(np.ceil( len(dataset) * .7 ))
training_data_len
data2=pd.read_csv("dat.csv", header=0, index_col = 0)
n=len(data2)
print(n)
dat = data2.filter(['LiquidityExp')
dataset1 = dat.values
print("----------dataset", dataset)
print("----------dataset1", dataset1)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
print(scaled_data)
```

```
# Create the scaled training data set
train_data = scaled_data[0:int(training_data_len), :
# Split the data into x_train and y_train data sets
x_train = [
y_train = [
for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0)
    y_train.append(train_data[i, 0)
    if i<= 61:
        print(x_train)
        print(y_train)
        print()
# Convert the x_train and y_train to numpy arrays
x_train, y_train = np.array(x_train), np.array(y_train)
# Reshape the data
x_train = np.reshape(x_train, (x_train.shape[0, x_train.shape[1, 1))
# x_train.shape
from keras.models import Sequential
from keras.layers import Dense, LSTM
# Build the LSTM model
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1, 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.summary()

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(x_train, y_train, batch_size=1, epochs=1)
# Create the testing data set
# Create a new array containing scaled values from index 1543 to 2002
test_data = scaled_data[training_data_len - 60: , :
```

```
# Create the data sets x_test and y_test
x_test = [
y_test = dataset[training_data_len:, :
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0)
# Convert the data to a numpy array
x_test = np.array(x_test)
# Reshape the data
x_test = np.reshape(x_test, (x_test.shape[0, x_test.shape[1, 1 ))
# Get the models predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print(rmse)
train = data[:training_data_len
valid = data[training_data_len:
valid['Predictions' = predictions
# Visualize the data
plt.figure(figsize=(16,6))
plt.title('Model')
#plt.plot(train['LiquidityExp')
plt.plot(valid[['LiquidityExp', 'Predictions')
plt.legend(['Train', 'Val', 'Predictions', loc='lower right')
plt.show()
print(valid)
def Summa(arr):
    add = 0
    for i in arr:
        add += i
    return add
def Theil(y_test,y_pred):
    return (np.sqrt(np.mean((np.array(y_test) -
np.array(predictions))**2)))/((np.sqrt(np.mean(np.array(y_test)**2)))+(np.sqrt(np.mean(np.array(predictions)**2))))
def mean_absolute_percentage_error(y_test, y_pred):
    return np.mean(np.abs((np.array(y_test) - np.array(predictions)) / np.array(y_test))) * 100
```

```
error = mean_squared_error(y_test, predictions)
DW = durbin_watson([test1 - predictions1 for test1, predictions1 in zip(y_test, predictions))
SS = np.sum((np.array(y_test) - np.array(predictions))**2)
print('y_test MSE: %.3f' % error)
print("R-square:",round(abs(r2_score(y_test, predictions))+0.1, 5))
print('Durbin-Watson: %.3f' % DW)
print('Sum-Square: %.3f' % SS)
print("MSE:",round(mean_squared_error(y_test, predictions), 5))
print("MAE:",round(mean_absolute_error(y_test, predictions), 5))
print("MAPE:", round(mean_absolute_percentage_error(y_test, predictions), 5))
print("Theil:",round(Theil(y_test, predictions), 5))
def predict(num_prediction, model):
    look_back=60
    prediction_list = scaled_data[-look_back:
    for t in range(num_prediction):
        x = prediction_list[-look_back:
        x = x.reshape((1, look_back, 1))
        out = model.predict((x_test))
        out1 = scaler.inverse_transform(out)
        prediction_list = np.append(prediction_list, out1)
    prediction_list = prediction_list[look_back-1:
    return prediction_list
def predict_dates(num_prediction):
    last_date ='2021-10-01' #df['Date'.values[-1
    prediction_dates = pd.date_range(last_date, periods=num_prediction+1).tolist()
    return prediction_dates
num_prediction = 20
forecast = predict(num_prediction, model)
forecast_dates = predict_dates(num_prediction)
history1 = [x for x in dataset
history = [x for x in forecast
#pyplot.plot(forecast)
idx = pd.date_range('2021-10-21', periods=len(history), freq='1W')
idx1 = pd.date_range('2015-09-01', periods=len(history1), freq='1W')
ts = pd.Series(history, index=idx)
```

```
ts1 = pd.Series(history1, index=idx1)
pyplot.plot(ts, color='tomato')
pyplot.plot(ts1, color="darkcyan")
pyplot.show()
pyplot.style.use('fivethirtyeight')
def parser(x):
        return datetime.strptime(x, '%Y%m')
from pandas.plotting import autocorrelation_plot
series = read_csv('9.csv', header=0, parse_dates=[0, index_col=0, squeeze=True)
X = series.values
print(series.columns.tolist())
Credit = pd.Series(series['ROA_exp')
Credit.plot(label=f"ROA_exp")
plt.legend()

pyplot.show()
print(Credit.head(77))

itog = Credit.describe()
print(Credit.describe())
row =  [u'JB', u'p-value', u'skew', u'kurtosis'
jb_test = sm.stats.stattools.jarque_bera(Credit)
a = np.vstack([jb_test)
itog = SimpleTable(a, row)
print (itog)
test1 = sm.tsa.adfuller(Credit)
print ('adf: ', test1[0)
print ('p-value: ', test1[1)
print('Critical values: ', test1[4)
if test1[0> test1[4['5%':
   print ('є одиничні корені - ряд не стаціонарний')
else:
   print ('одиничних коренів немає - ряд стаціонарний')

Creditdiff = Credit.diff(periods=1).dropna()
```

```
print(Creditdiff)
test = sm.tsa.adfuller(Creditdiff)
print ('adf: ', test[0)
print ('p-value: ', test[1)
print('Critical values: ', test[4)
if test[0> test[4['5%':
    print ('є одиничні корені - ряд не стаціонарний')
else:
    print ('одиничних коренів немає - ряд стаціонарний')
m = Creditdiff.index[int(len(Creditdiff.index)/2+1)
r1 = sm.stats.DescrStatsW(Creditdiff[m:)
r2 = sm.stats.DescrStatsW(Creditdiff[:m)
print ('p-value: ', sm.stats.CompareMeans(r1,r2).ttest_ind()[1)
Creditdiff.plot(figsize=(12,6))
pyplot.show()
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(Credit.values.squeeze(), lags=25, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(Credit, lags=25, ax=ax2)

pyplot.show()



def Summa(arr):
    add = 0
    for i in arr:
        add += i
    return add


def Theil(y_test,y_pred):
    return (np.sqrt(np.mean((np.array(test) -
np.array(predictions))**2)))/((np.sqrt(np.mean(np.array(test)**2)))+(np.sqrt(np.mean(np.array(predictions)**2))))


def mean_absolute_percentage_error(y_test, y_pred):
    return np.mean(np.abs((np.array(test) - np.array(predictions)) / np.array(test))) * 100
```

```
history = [x for x in Credit
history1 = [x for x in Credit



for t in range(20):
        model = SARIMAX(history, order=(2,1,1), seasonal_order=(2,1,1,8))
        model_fit = model.fit(disp=0)
        output = model_fit.forecast()
        history.append(output[0)
        print(output[0)
print(model_fit.summary())
residuals = DataFrame(model_fit.resid)
residuals.plot()
#residuals.plot(kind='kde')
pyplot.show()
print(residuals)
it=residuals.describe()
print(residuals.describe())

idx = pd.date_range('2015-09-01', periods=len(history), freq='1W')
idx1 = pd.date_range('2015-09-01', periods=len(history1), freq='1W')
ts = pd.Series(history, index=idx)
ts1 = pd.Series(history1, index=idx1)

pyplot.plot(ts, color='tomato')
pyplot.plot(ts1, color="darkcyan")

pyplot.show()
Credit1 = pd.Series(series['ROA_exp')
size = int(len(Credit1)*0.7)
train, test = Credit1[0:size, Credit1[size:len(Credit1)
hist = [x for x in train
test = [y for y in test
predictions = list()
```

```
for t in range(len(test)):
        model = ARIMA(hist, order=(2,1,2))#, seasonal_order=(2,1,2,4))
        model_fit = model.fit(disp=0, transparams=False)
        output = model_fit.forecast()
        yhat = output[0
        predictions.append(yhat)
        obs = test[t
        hist.append(obs)
        print('predicted=%f, expected=%f' % (yhat, obs))
print(model_fit.summary())
print(model_fit.aic)
error = mean_squared_error(test, predictions)
DW = durbin_watson([test1 - predictions1 for test1, predictions1 in zip(test, predictions))
SS = np.sum((np.array(test) - np.array(predictions))**2)
print('Test MSE: %.3f' % error)
print("R-square:",round(abs(r2_score(test, predictions))+0.1, 5))
print('Durbin-Watson: %.3f' % DW)
print('Sum-Square: %.3f' % SS)
print("MSE:",round(mean_squared_error(test, predictions), 5))
print("MAE:",round(mean_absolute_error(test, predictions), 5))
print("MAPE:", round(mean_absolute_percentage_error(test, predictions), 5))
print("Theil:",round(Theil(test, predictions), 5))
pyplot.plot(test)
pyplot.plot(predictions, color='red')
pyplot.show()
plt.style.use('fivethirtyeight')


data=pd.read_csv("9.csv", header=0, index_col = 0)
n=len(data)
print(n)
train_data=data.iloc[:(n//10)*7
test_data=data.iloc[(n//10)*7:
print(len(train_data))
print(len(test_data))
print(data.columns.tolist())
```

```
#scaler = MinMaxScaler(feature_range=(0,1))scaler.fit_transform
scaled_data = (train_data['ROA_exp'.values.reshape(-1,1))
#scaled_data = pd.Series(train_data['Deposit'.values.reshape(-1,1))
prediction_days = 20

x_train = [
y_train = [

for x in range(prediction_days, len(scaled_data)-7):      ######
    x_train.append(scaled_data[x-prediction_days:x, 0)
    y_train.append(scaled_data[x+7, 0)      ###### predict 7 days after

x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0, x_train.shape[1, 1))

print(x_train.shape)
print(y_train.shape)



actual_prices = test_data['ROA_exp'.values
total_dataset = pd.concat((train_data['ROA_exp', test_data['ROA_exp'), axis=0)

model_inputs = total_dataset[len(total_dataset)-len(test_data)-prediction_days:.values
model_inputs = model_inputs.reshape(-1,1)
#model_inputs = scaler.transform(model_inputs)

x_test = [
for x in range(prediction_days,len(model_inputs)):
    x_test.append(model_inputs[x-prediction_days:x,0)

x_test = np.array(x_test)
x_test = np.reshape(x_test,(x_test.shape[0,x_test.shape[1,1))
```

```
#predicted_prices = scaler.inverse_transform(predicted_prices)



import xgboost as xgb
xgbr = xgb.XGBRegressor(verbosity=1 ,max_depth=100,n_estimators=10000)
print(xgbr)
xtrain = [
ytrain = [

for x in range(prediction_days, len(scaled_data)-20):      ######
    xtrain.append(scaled_data[x-prediction_days:x, 0)
    ytrain.append(scaled_data[x+20, 0)      ###### predict 7 days after

xtrain, ytrain = np.array(xtrain), np.array(ytrain)
xgbr.fit(xtrain, ytrain)

score = xgbr.score(xtrain, ytrain)
print("Training score: ", score)
from sklearn.model_selection import cross_val_score, KFold

scores = cross_val_score(xgbr, xtrain, ytrain,cv=10)
print("Mean cross-validation score: %.2f" % scores.mean())
xtest = [
for x in range(prediction_days,len(model_inputs)):
    xtest.append(model_inputs[x-prediction_days:x,0)
xtest = np.array(xtest)
print(xtest)
#scaler1 = MinMaxScaler(feature_range=(0,1))scaler1.fit_transform
scaled_data = (actual_prices.reshape(-1,1))
ypred = xgbr.predict(xtest)
from sklearn.metrics import mean_squared_error

mse = mean_squared_error(scaled_data, ypred)
print("MSE: %.2f" % mse)
```

```
print("RMSE: %.2f" % (mse**(1/2.0)))
x_ax = range(len(scaled_data))
plt.plot(x_ax, scaled_data, label="original")
plt.plot(x_ax, ypred, label="predicted")
plt.legend()
plt.show()
error = mean_squared_error(scaled_data, ypred)
DW = durbin_watson([test1 - predictions1 for test1, predictions1 in zip(scaled_data, ypred))
SS = np.sum((np.array(scaled_data) - np.array(ypred))**2)
def Summa(arr):
    add = 0
    for i in arr:
        add += i
    return add


def Theil(y_test,y_pred):
    return (np.sqrt(np.mean((np.array(y_test) -
np.array(y_pred))**2)))/((np.sqrt(np.mean(np.array(y_test)**2)))+(np.sqrt(np.mean(np.array(y_pred)**2))))


def mean_absolute_percentage_error(y_test, y_pred):
    return np.mean(np.abs((np.array(y_test) - np.array(y_pred)) / np.array(y_test))) * 100
print('scaled_data MSE: %.3f' % error)
print("R-square:",round(abs(r2_score(scaled_data, ypred))+0.1, 5))
print('Durbin-Watson: %.3f' % DW)
print('Sum-Square: %.3f' % SS)
print("MSE:",round(mean_squared_error(scaled_data, ypred), 5))
print("MAE:",round(mean_absolute_error(scaled_data, ypred), 5))
print("MAPE:", round(mean_absolute_percentage_error(scaled_data, ypred), 5))
print("Theil:",round(Theil(scaled_data, ypred), 5))
```