

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

На правах рукопису  
УДК 303.732.4

До захисту допущено  
Завідувач кафедри ММСА  
\_\_\_\_\_ Оксана ТИМОЩУК  
«\_\_\_» \_\_\_\_\_ 2024 р.

**Магістерська дисертація**  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Системний аналіз фінансового ринку»  
зі спеціальності 124 «Системний аналіз»  
на тему: «Підхід щодо моделювання вартості акцій по закриттю біржового  
ринку із використанням нейронних мереж»

Виконав:

Студент 2 курсу, групи КА-32мп  
Гирила Степан Ігорович \_\_\_\_\_

Науковий керівник:

Старший викладач кафедри ММСА, к.т.н.,  
Савастьянов Володимир Володимирович \_\_\_\_\_

Рецензент:

Старший науковий співробітник ІТГПІ,  
доцент, д.т.н.,  
Терентьев Олександр Миколайович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць інших  
авторів без відповідних посилань  
Студент: \_\_\_\_\_

Київ - 2024

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

Рівень вищої освіти – другий (магістерський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійною програмою «Системний аналіз фінансового ринку»

ЗАТВЕРДЖУЮ

Завідувач кафедри ММСА

\_\_\_\_\_ Оксана ТИМОЩУК

«\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Гирилі Степану Ігоровичу**

1. Тема дисертації: «Підхід щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж», науковий керівник дисертації Савастьянов Володимир Володимирович, старший викладач кафедри ММСА, к.т.н, затверджені наказом по університету від «07» листопада 2024 р. № 5001-с
2. Строк подання студентом дисертації: 02.12.2024.
3. Об'єкт дослідження: акції на біржовому ринку, підходи до їх прогнозування.
4. Предмет дослідження: методи та моделі машинного навчання для моделювання вартості акцій, визначення залежності цих методів від волативності вхідних даних.
5. Перелік завдань, які потрібно зробити:
  - 1) проаналізувати та узагальнити основні терміни з теми;
  - 2) проаналізувати етапи побудови моделей прогнозування;
  - 3) описати принцип роботи моделей та підходи до оцінювання їх адекватності;
  - 4) здійснити та проаналізувати результати обчислювальних експериментів;
  - 5) розробити стартап-проект виведення на ринок результатів дослідження.
6. Перелік графічного (ілюстративного матеріалу): презентація
7. Орієнтовний перелік публікацій:

Гирила С.І., Савастьянов В.В. Підхід щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж. Системні науки та інформатика: збірник доповідей III Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 25 – 29 листопада 2024 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024, стор. 8.

## 8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання: 02.09.2024

## Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Затвердження теми магістерської дисертації та огляд технічної літератури	02.09.2024-09.09.2024	Виконано
2	Формулювання об'єкта, предмета, мети, актуальності та практичної значущості результатів	09.09.2024-16.09.2024	Виконано
3	Перший розділ. Актуальність теми дослідження та огляд предметної області	16.09.2024-30.09.2024	Виконано
4	Другий розділ. Використання нейронних мереж для прогнозування цін акцій	30.09.2024-07.10.2024	Виконано
5	Визначення набору даних, його початкова підготовка	07.10.2024-14.10.2024	Виконано
6	Проведення експериментів, та аналіз отриманих результатів	14.10.2024-28.10.2024	Виконано
7	Третій розділ. Огляд програмного продукту та аналіз отриманих результатів	28.10.2024-11.11.2024	Виконано
8	Четвертий розділ. Розробка власного стартап-проекту	11.11.2024-18.11.2024	Виконано
9	Висновки по роботі та перспективи подальших досліджень	18.11.2024-25.11.2024	Виконано
10	Оформлення магістерської дисертації	25.11.2024-02.12.2024	Виконано

Студент \_\_\_\_\_ Степан ГИРИЛА

Науковий керівник дисертації \_\_\_\_\_ Володимир САВАСТЬЯНОВ

## РЕФЕРАТ

Загальний обсяг роботи: 108 с., 39 рис., 21 табл., 2 дод., 33 джерела.

ВАРТІСТЬ АКЦІЙ, ПРОГНОЗУВАННЯ, НЕЙРОННІ МЕРЕЖІ, ФОНДОВИЙ РИНОК, RANDOM FOREST, DNN, LSTM.

Об'єкт дослідження: відкриті дані про торги акціями на фондовому ринку.

Мета дослідження: дослідження ефективності існуючих моделей для прогнозування вартості акцій на фондовому ринку, виявлення їх сильних та слабких сторін у контексті волативності ринку.

Актуальність дослідження: прогнозування вартості акцій на фоні зростаючої волативності фондових ринків є важливою темою для інвесторів та трейдерів.

Дослідження спрямоване на аналіз ефективності популярних методів машинного навчання, включно з алгоритмами глибокого навчання, у контексті підвищення точності прогнозів вартості акцій за умов зростаючої волативності ринків. У його рамках передбачено вивчення сучасних моделей прогнозування вартості акцій із оцінкою їх результативності за різного рівня волативності вхідних даних. Також увагу буде приділено виявленню існуючих вад і обмежень поширених методів прогнозування.

У даній роботі буде представлена програмна реалізація багат шарової нейронної мережі, мережі довгої короткочасної пам'яті та методу Random Forest. Для реалізації було обрано мову програмування Python (версія 3.9.12) та середу розробки Jupyter Notebook.

## ABSTRACT

Total amount of work: 108 p., 39 figures, 21 tables, 2 appendices, 33 sources.

STOCK PRICE, FORECASTING, NEURAL NETWORKS, STOCK MARKET, RANDOM FOREST, DNN, LSTM.

Object of research: open data on stock trading in the stock market.

Purpose of the study: to study the effectiveness of existing models for predicting the value of shares in the stock market, to identify their strengths and weaknesses in the context of market volatility.

The relevance of the study: forecasting the value of shares against the background of growing volatility of stock markets is an important topic for investors and traders.

The study aims to analyze the effectiveness of popular machine learning methods, including deep learning algorithms, in improving the accuracy of stock price forecasts in the context of growing market volatility. The study envisages examining modern stock price forecasting models with an assessment of their performance at different levels of input volatility. Attention will also be paid to identifying the existing shortcomings and limitations of common forecasting methods.

In this paper, will be present a software implementation of a multilayer neural network, a long short-term memory network, and the Random Forest method. For the implementation, we chose the Python programming language (version 3.9.12) and the Jupyter Notebook development environment.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ.....</b>	<b>8</b>
<b>ВСТУП.....</b>	<b>9</b>
<b>РОЗДІЛ 1 АКТУАЛЬНІСТЬ ТЕМИ ДОСЛІДЖЕННЯ ТА ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ .....</b>	<b>11</b>
1.1 Актуальність теми дослідження.....	11
1.2 Підходи щодо аналізу цін акцій.....	12
1.2.1 Технічний аналіз.....	13
1.2.2 Фундаментальний аналіз .....	14
1.3 Аналіз попередніх досліджень .....	16
1.4 Сучасні інструменти, які використовуються для прогнозування .....	18
1.5 Висновки до розділу 1 .....	19
<b>РОЗДІЛ 2 ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ ЦІН АКЦІЙ .....</b>	<b>21</b>
2.1 Вступ до розділу 2 .....	21
2.2 Створення нейронних мереж.....	22
2.3 Способи оптимізації кількості кластерів .....	25
2.3.1 Метод ліктя .....	25
2.3.2 Метод Silhouette score .....	27
2.4 Функції активації .....	28
2.4.1 Функція активації ReLU .....	28
2.4.2 Функція активації ELU .....	30
2.4.4 Функція активації Softmax .....	32
2.5 Багатошарова нейронна мережа.....	33
2.6 Мережа довгої короткочасної пам'яті (LSTM) .....	35
2.7 Random forest.....	38
2.8 Методи візуалізації вхідних даних .....	40
2.9 Схема підходу .....	42
2.10 Висновки до розділу 2.....	46

<b>РОЗДІЛ 3 ОГЛЯД ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ .....</b>	<b>47</b>
3.1 Вступ до розділу 3 .....	47
3.2 Візуалізація вхідних даних .....	47
3.3 Відсоткова зміна цін на акції .....	51
3.5 Реалізація мережі довгої короткочасної пам'яті .....	57
3.6 Реалізація алгоритму Random Forest .....	60
3.7 Реалізація рандомного торгового механізму .....	62
3.8 Висновки до розділу 3 .....	64
<b>РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ .....</b>	<b>66</b>
4.1 План розробки стартапу та масштабування його на ринок .....	66
4.2 Опис ідеї стартап-проекту .....	68
4.3 Технологічний аудит ідеї проекту .....	69
4.4 Аналіз ринкових можливостей запуску стартап-проекту .....	71
4.5 Розроблення ринкової стратегії стартап-проекту .....	81
4.6 Розроблення маркетингової програми стартап-проекту .....	84
4.7 Висновки до розділу 4 .....	86
<b>ВИСНОВКИ .....</b>	<b>87</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....</b>	<b>88</b>
<b>ДОДАТОК А ЛІСТИНГ ПРОГРАМИ .....</b>	<b>92</b>
<b>ДОДАТОК Б ДОПОВІДЬ У ВИГЛЯДІ ПРЕЗЕНТАЦІЇ .....</b>	<b>102</b>

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ**

EPS – Earnings per share

ROE – Return on equity

ARMA – Autoregressive-moving average

ReLU – Rectified linear unit

ELU – Exponential Linear Unit

TPR – True Positive Rate

FPR – False Positive Rate

TNR – True Negative Rate

FNR – False Negative Rate

RMSE – Root Mean Squared Error

DNN – Deep Neural Network

RNN – recurrent neural network

LSTM – Long Short-Term Memory



## ВСТУП

На сьогоднішній час фінансовий ринок стає все складнішим та динамічнішим. Глобалізація, яка притаманна сучасному світу, створює численні фактори впливу, починаючи від зміни економічних показників, закінчуючи глобальними подіями. Наприклад, стихійне лихо на одному кінці планети може вплинути на стан ринку на іншому кінці, викликавши різке падіння, або навпаки зростання певних секторів фінансового ринку. У такому складному середовищі можливість передбачити зміни у ціні акцій дозволяють трейдерам та інвесторам приймати зважені рішення.

Метою даної магістерської роботи є дослідження ефективності існуючих моделей для прогнозування вартості акцій на фондовому ринку, виявлення їх сильних та слабких сторін у контексті волативності ринку. Із зростанням фондових ринків зростає попит на методи, які здатні прогнозувати зміну цін на акції. Проте, таке зростання супроводжується із зростанням волативності цін на акції, що впливає на точність цих моделей. Тому це дослідження має на меті внести певне роз'яснення, які методи показують найкращі результати при різній волативності.

Завдання дослідження:

- 1) дослідити ефективність популярних методів машинного навчання, включаючи алгоритми глибокого навчання, для підвищення точності прогнозів вартості акцій при зростаючій волативності ринків;
- 2) проаналізувати популярні моделі прогнозування вартості акцій, оцінити їх ефективність за різної волативності вхідних даних;
- 3) визначити існуючі вади та обмеження у популярних методах прогнозування вартості акцій.

Магістерська дисертація складається із вступу, чотирьох розділів, висновків, списку використаної літератури та одного додатку. В першому

розділі розглянуто актуальність теми дослідження, основні підходи щодо аналізу цін на акції, такі як технічний та фундаментальний аналіз. Розглянуто існуючі дослідження в даній сфері та інструменти, які використовуються для прогнозування цін на акції. В другому розділі розглянуто популярні моделі та методи прогнозування та функції активації. Також звернено увагу на методи кластеризації. В третьому розділі розглянуто реалізацію таких нейронних мереж як багат шарова нейронна мережа, мережа довгої-короткочасної пам'яті та методу Random Forest. Проведено аналіз отриманих результатів за метриками TPR, FPR, TNR, FNR та RMSE при різній волативності вхідних даних. У четвертому розділі проведено дослідження щодо доцільності розробки стартап-проекту та його комерційного використання на основі ідей, описаних в магістерській дисертації. Додаток А містить лістинг програми з реалізацією моделей та методів прогнозування.

## РОЗДІЛ 1 АКТУАЛЬНІСТЬ ТЕМИ ДОСЛІДЖЕННЯ ТА ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Актуальність теми дослідження

В умовах постійного зростання темпів економічного розвитку та глобалізації фінансові ринки набувають все більшої складності та динамічності. Для учасників ринку надзвичайно важливо не лише усвідомлювати поточні тенденції, але й мати можливість прогнозувати майбутні події. Це є необхідним для прийняття обґрунтованих управлінських рішень, розробки інвестиційних стратегій та зменшення ризиків.

Прогнозування вартості акцій на момент закриття є однією з основних задач у фінансовій аналітиці, оскільки точність прогнозів безпосередньо впливає на ефективність інвестиційних рішень. Однак складність фінансових даних, нелінійні залежності між показниками та висока волатильність ринків ускладнюють процес моделювання. У зв'язку з цим традиційні методи статистичного аналізу часто не забезпечують необхідної точності, що робить використання сучасних підходів, таких як нейронні мережі, особливо актуальним.

Актуальність теми ґрунтується на низці ключових аспектів, які визначають необхідність такого дослідження.

По-перше, зростаюча волативність фінансових ринків та обсяги даних вимагають нових інструментів для аналізу. Традиційні статистичні методи часто не враховують складні нелінійні залежності та великі обсяги історичних даних, що робить їх недостатньо ефективними у прогнозуванні вартості акцій. Нейронні мережі, зокрема рекурентні та глибокі, здатні ефективно моделювати складні взаємозв'язки між змінними, враховуючи як числові, так і категоріальні дані. Це відкриває можливості не лише для створення більш

точних прогностичних моделей, але й для виявлення аномалій, розкриття прихованих закономірностей та адаптації моделей до різних ринкових умов.

По-друге, глобалізація фінансових ринків та їх взаємозалежність підвищують вимоги до якості прогнозування. Відсутність універсальних моделей для прогнозування фінансових показників вимагає розробки специфічних підходів, які враховують особливості конкретного ринку або фінансового інструменту. Неправильний вибір моделі або її недостатня адаптація до ринкових умов можуть призвести до значних втрат для інвесторів, компаній та інших учасників ринку.

По-третє, розробка методів прогнозування на основі нейронних мереж є важливим завданням як з наукової, так і з практичної точки зору. З наукової перспективи це сприяє розвитку методів інтелектуального аналізу даних та глибокого навчання. З практичного боку, це відкриває можливості для прийняття ефективних управлінських рішень, зменшення ризиків та підвищення конкурентоспроможності на фінансових ринках.

## 1.2 Підходи щодо аналізу цін акцій

Для аналізу цін акцій існують кілька ключових підходів, кожен з яких базується на різних методах та підходах до прогнозування фінансових ринків. Одним з найстаріших таких методів є технічний аналіз.

### 1.2.1 Технічний аналіз

Технічний аналіз – це метод оцінки статистичних тенденцій у торговельній діяльності, як правило, пов'язаних з рухом цін і обсягів. Він використовується для виявлення торгових та інвестиційних можливостей.

Технічний аналіз використовується для аналізу впливу попиту та пропозиції на зміни в ціні цінного паперу, обсязі та передбачуваний волатильності. Він припускає, що минула торгова активність і зміни ціни цінного паперу можуть бути цінними індикаторами майбутнього руху ціни цінного паперу в поєднанні з відповідними правилами інвестування або торгівлі.

Технічний аналіз використовує різноманітні графічні інструменти для створення короткострокових торгових сигналів. Ці інструменти також сприяють покращенню оцінки сили або слабкості цінного паперу в порівнянні з більш широким ринком або його секторами. За рахунок отриманої інформації аналітики здатні покращити загальну оцінку цінних паперів.

Сучасний технічний аналіз, яким ми його знаємо, вперше був представлений Чарльзом Доу наприкінці 1800-х років під назвою «Теорія Доу». Відомі дослідники, такі як Вільям П. Гамільтон, Роберт Реа, Едсон Гулд та Джон Мейджі, також внесли значний вклад у розвиток концепцій цієї теорії [1]. Сьогодні технічний аналіз еволюціонував і включає сотні моделей і сигналів, які були розроблені в результаті багаторічних досліджень.

Професійні трейдери та аналітики часто прибігають до поєднання технічного аналізу з іншими методами дослідження. Роздрібні трейдери можуть ухвалювати рішення, спираючись лише на цінові графіки та відповідну статистику цінних паперів. Проте, фондові аналітики зазвичай не обмежуються тільки технічним аналізом у своїй роботі.

Технічний аналіз може бути застосований до будь-якого цінного паперу, для якого є історичні торгові дані. Це стосується акцій, ф'ючерсів,

товарів, цінних паперів з фіксованим доходом, валют тощо. Насправді, технічний аналіз особливо популярний на товарних і валютних ринках, де трейдери зосереджуються на короткострокових коливаннях цін.

Метою технічного аналізу є прогнозування цінових рухів практично будь-якого торгового інструменту, який зазвичай підлягає впливу попиту та пропозиції. Деякі вважають, що технічний аналіз є відображенням сил попиту і пропозиції, які проявляються в русі ринкової ціни цінного паперу [2].

### 1.2.2 Фундаментальний аналіз

Фундаментальний аналіз полягає у вивченні фінансової звітності компанії та загальних економічних показників з метою визначення її внутрішньої вартості. Результати цього аналізу мають надати реальне уявлення про вартість інвестиції, враховуючи фінансовий стан компанії, ринкові умови та економічну ситуацію. Інвестори здійснюють фундаментальний аналіз, щоб зрозуміти, чи варто вкладати кошти в компанію, спираючись на її поточну та прогнозовану вартість.

Під час проведення фундаментального аналізу досліджується зростання доходів компанії, її прибутковість та конкурентні переваги в галузі. Також оцінюються макроекономічні фактори, такі як загальний стан економіки та попит на продукцію чи послуги компанії. Не менш важливим є аналіз ефективності управлінської команди. Кваліфіковані та досвідчені керівники здатні подолати труднощі та скористатися можливостями, що сприяє зростанню компанії та підвищенню її вартості.

Такий підхід до аналізу ринку часто дозволяє заглянути за межі настроїв інвесторів і маркетингових стратегій компанії, щоб оцінити її потенціал для досягнення довгострокового успіху. Використовуючи

фундаментальний аналіз, ви можете з'ясувати, чи є ринкова ціна цінного паперу завищеною або заниженою.

Фундаментальний аналіз зазвичай починається з вивчення фінансової звітності компанії, яка включає звіт про прибутки і збитки, баланс та звіт про рух грошових коштів. Інформація, що міститься в цих документах, може бути використана для оцінки фінансового стану компанії та її внутрішньої вартості. Основні коефіцієнти, отримані з цих фінансових звітів, включають співвідношення ціни до прибутку (P/E), прибуток на акцію (EPS), рентабельність власного капіталу (ROE) та співвідношення боргу до власного капіталу (D/E). Фундаментальні аналітики використовують ці показники для визначення, чи є акції недооціненими або переоціненими в порівнянні з ринковою ціною та конкурентами [3].

Фундаментальний аналіз також включає в себе оцінку валового внутрішнього продукту, інфляції, рівня безробіття, а також тенденцій у певних галузях і конкурентного середовища компанії. Таким чином, він аналізує не лише ринкову діяльність компанії, а й її роль як виробника товарів і послуг. Для цього важливо врахувати загальний стан економіки, результати роботи сектора та позицію компанії в індустрії, щоб визначити її вартість і спрогнозувати майбутні показники. Цей підхід не обмежується лише економічними та фінансовими даними, а також включає аналіз бізнес-моделі, ефективності управлінських процесів, впізнаваності бренду, а також потенціалу для зростання і прибутковості [4].

В результаті, метою фундаментального аналізу є визначення вартості компанії, яку можна використовувати при купівлі, утриманні або продажу акцій. Це вимагає глибокого розуміння фінансової звітності та стратегічного бачення того, як зовнішні фактори можуть вплинути на майбутні прибутки та ринкову позицію компанії.

### 1.3 Аналіз попередніх досліджень

Прогнозування біржових ринків та фінансових даних є важливою темою для фінансових аналітиків та інвесторів уже багато років. Одною з перших значних робіт у даному напрямку стала робота над статистичним аналізом часових рядів. У 1970 році статистики Джордж Бокс та Гвілім Дженкінс запропонували те, що стало відомим як метод Бокса-Дженкінса (англ. Box-Jenkins method) для побудови моделі будь-якого типу часового ряду. Підхід починається з припущення, що процес, який породив часовий ряд, може бути апроксимований за допомогою моделі, якщо він є стаціонарним.

Він складається з чотирьох кроків.

1. Ідентифікація – оцінюється, чи є часовий ряд стаціонарним, і якщо ні, то скільки різниць потрібно для того, щоб зробити його стаціонарним. Потім генеруються різницеві дані для використання в діагностичних графіках. Визначаються параметри моделі ARMA (autoregressive-moving average) для даних з автокореляцією та частковою автокореляцією.
2. Оцінка – використовуються дані для навчання параметрів моделі (тобто коефіцієнтів).
3. Діагностична перевірка – оцінюються підібрані моделі в контексті наявних даних і перевіряється, де модель може бути покращена. Зокрема, це включає розрахунок залишкових помилок.
4. Прогнозування – після проходження попередніх етапів отримується модель, з-за допомогою якої можна починати прогнозування значень [2].

Даний метод був опублікований у книзі «Time Series Analysis: Forecasting and Control» [5].

У 2000-х роках модель експоненційного згладжування (ETS), розроблена Робом Гайдманом та його командою, стала альтернативним



методом для аналізу часових рядів. Ця модель акцентує увагу на обробці даних, що містять тренди та сезонні коливання. Цей підхід мав декілька публікацій, після чого він був розширений та викладений у книзі Гайдмана, виданій у 2008 році [6].

З розвитком комп'ютерних технологій методи машинного навчання отримали значне поширення. Дослідження Джеймса Граета, Данієли Віттен, Тревора Гасті та Роберта Тібширані, представлені у 2013 році в книзі «An Introduction to Statistical Learning» [7], продемонстрували, що методи машинного навчання, такі як лінійна та поліноміальна регресія, можуть бути ефективними для аналізу великих обсягів даних. Вони здатні виявляти залежності між змінними та прогнозувати фінансові показники на основі історичних даних [8].

Мережі глибокого навчання, які здобули популярність у 2010-х роках, надають можливість аналізувати складні часові ряди з нелінійними залежностями. Ці моделі ефективно використовуються для виявлення аномалій, складних взаємозв'язків та підвищення точності прогнозів.

Попередні дослідження також демонструють, що ефективність моделі залежить від характеру вхідних даних. Наприклад, модель ARIMA підходить для даних зі стаціонарними характеристиками, тоді як глибокі нейронні мережі краще справляються з великими нестабільними наборами даних.

Дослідження, викладене в цій роботі, ставить за мету вивчення точності представлених моделей у контексті волативності вхідних даних при моделюванні вартості акцій по закриттю біржового ринку. Це сприятиме фінансовим аналітикам і трейдерам у прийнятті обґрунтованих рішень на основі отриманих прогнозів.

## 1.4 Сучасні інструменти, які використовуються для прогнозування

На даний момент часу існують різноманітні інструменти для моделювання вартості акцій. Ці інструменти поєднують алгоритми машинного навчання, статистичні методи аналізу та потужні середовища розробки, що забезпечують ефективну обробку даних.

На сьогоднішній час мова програмування Python займає одне з перших місць у сфері машинного навчання та аналізу даних. Вона стала популярною за рахунок простоти використання, гнучкості та багатого вибору бібліотек для обробки, аналізу та візуалізації даних. Наприклад, бібліотека Scikit-Learn є відмінним інструментом для створення моделей регресії та класифікації, тоді як TensorFlow і Keras дозволяють розробляти складні нейронні мережі, зокрема рекурентні, які підходять для аналізу часових рядів. Крім того, бібліотеки NumPy і Pandas є незамінними для роботи з числовими даними та табличними структурами, що є важливим у фінансових дослідженнях.

Якщо потрібна альтернатива для Python у даній сфері, варто звернути увагу на мову програмування R. Вона добре підходить для прогнозування, зокрема завдяки своїй сильній орієнтації на статистичний аналіз. Використовуючи пакети, такі як `forecast` і `tseries`, можна ефективно аналізувати часові ряди, виявляти закономірності та створювати моделі, які враховують сезонність та інші складові фінансових даних. Крім того, R славиться своїми потужними засобами візуалізації. Наприклад, бібліотека `ggplot2` дозволяє створювати професійні графіки, які роблять результати аналізу зрозумілими для широкої аудиторії.

Обираючи середовища обробки варто звернути увагу на інтерактивні середовища, такі як Jupyter Notebook. Правильно обране середовище також відіграє важливу роль у розробці моделей і презентації результатів. Вони дозволяють поєднувати код, текстові пояснення та візуалізацію в одному документі, що є зручним для дослідників, які прагнуть швидко

експериментувати з алгоритмами, аналізувати дані в реальному часі та документувати свої знахідки в інтегрованій формі.

Альтернативою для інтерактивних середовищ можуть бути хмарні сервіси, які забезпечують доступ до потужних обчислювальних ресурсів. Наприклад, Google Colab дозволяє безкоштовно використовувати Python у середовищі з підтримкою GPU. Такі платформи, як Amazon SageMaker і Microsoft Azure Machine Learning, пропонують професійні рішення для навчання моделей, їх інтеграції в бізнес-процеси та розгортання в реальних умовах.

Джерела фінансових даних, такі як Yahoo Finance API, Alpha Vantage та Quandl, надають доступ до історичних та актуальних ринкових показників, які є основою для створення точних моделей. Ці платформи дозволяють автоматизувати завантаження великих обсягів інформації, що значно спрощує роботу аналітиків і дослідників.

Поєднання потужних мов програмування, інтерактивних середовищ, хмарних платформ і спеціалізованих бібліотек створює ідеальні умови для аналізу фінансових даних. Завдяки їм можна ефективно розв'язувати складні завдання, зокрема прогнозувати ринкову динаміку та знижувати фінансові ризики.

## 1.5 Висновки до розділу 1

У даному розділі звернено увагу на значущості та актуальності прогнозування цін на акції, використовуючи сучасні підходи до аналізу фінансових ринків. Також розглянуто актуальність теми дослідження.

Технічний та фундаментальний аналіз, які довели свою ефективність у минулому, залишаються основою для оцінки ринкової динаміки. Однак складність сучасних фінансових даних і високий рівень волативності

вимагають впровадження інноваційних інструментів, таких як алгоритми машинного навчання, зокрема нейронні мережі.

Попередні дослідження підтверджують ефективність традиційних методів аналізу часових рядів. Водночас, поява методів глибокого навчання відкрила нові можливості для роботи з нелінійними залежностями та великими обсягами даних. Сучасні інструменти, такі як Python, R, Jupyter Notebook, хмарні платформи та API для доступу до фінансових даних, створюють основу для розробки ефективних моделей прогнозування.

Таким чином, використання поєднання традиційних та сучасних підходів є ключовим для розробки точних моделей, що враховують специфіку ринкових даних та дозволяють приймати обґрунтовані фінансові рішення. Це сприяє підвищенню конкурентоспроможності аналітиків та інвесторів в умовах глобалізації фінансових ринків.

## РОЗДІЛ 2 ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ ЦІН АКЦІЙ

### 2.1 Вступ до розділу 2

Нейронні мережі, які являють собою основні компоненти глибокого навчання, завдячують своїм концептуальним основам складним біологічним мережам нейронів у людському мозку. Ця концепція почалася з фундаментальної аналогії, проведення паралелей між біологічними нейронами та обчислювальними мережами.

Ця аналогія зосереджена навколо мозку, який складається з приблизно 100 мільярдів нейронів. Кожен нейрон підтримує близько 7000 синаптичних зв'язків з іншими нейронами, створюючи складну нейронну мережу, яка лежить в основі когнітивних процесів людини та прийняття рішень.

Індивідуально біологічний нейрон функціонує завдяки серії простих електрохімічних процесів. Він отримує сигнали від інших нейронів через свої дендрити. Коли ці вхідні сигнали досягають певного рівня (заздалегідь визначеного порогу), нейрон вмикається і надсилає електрохімічний сигнал по своєму аксону. Це, в свою чергу, впливає на нейрони, підключені до терміналів його аксону. Схематичне зображення біологічного нейрона можна побачити на рисунку 2.1. Важливо зазначити, що реакція нейрона подібна до двійкового перемикача: він або спрацьовує (активується), або не реагує, без жодних проміжних станів [9].

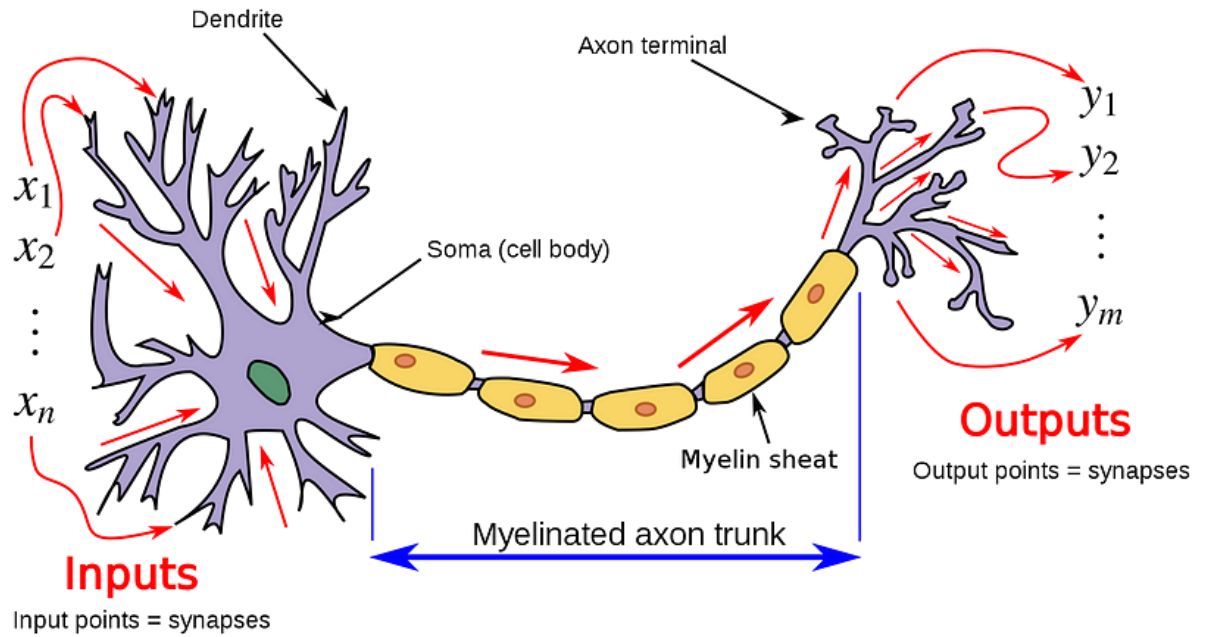


Рисунок 2.1 – Схема біологічного нейрона

## 2.2 Створення нейронних мереж

Фундаментальним будівельним блоком у розвитку нейронних мереж став нейрон Мак-Каллока-Піттса (англ. McCulloch-Pitts (MP) neuron).

Концепція MP-нейрона бере свій початок з роботи Уоррена Мак-Каллока та Волтера Піттса у 1943 році. Мак-Каллок, нейробіолог, і Піттс, логік, співпрацювали, щоб розробити математичну модель роботи нейронів у мозку. Їхня фундаментальна праця «A Logical Calculus of Ideas Immanent in Nervous Activity» заклала основу теорії нейронних мереж [10].

MP-нейрон – це спрощена модель біологічного нейрона. Він отримує двійкові входи (зазвичай 0 або 1) і видає двійковий вихід на основі порогового критерію. Модель складається з двох основних компонентів: входів і порогу. Булева логіка відіграє фундаментальну роль у розумінні функціонування нейронів Мак-Каллока-Піттса (MP). Розглянемо MP-нейрон детальніше.

1. Бінарні входи – у булевій логіці входи – це двійкові змінні, які можуть приймати одне з двох значень: true (1) або false (0). Аналогічно, в контексті МР-нейронів, входи - це двійкові сигнали, які представляють наявність (1) або відсутність (0) активації від підключених нейронів або зовнішніх джерел.
2. Порогова логіка – процес прийняття рішень МР-нейроном схожий на логічну операцію. Він підсумовує двійкові входи, які отримує, і порівнює результат із заздалегідь визначеним порогом. Якщо сума перевищує поріг, нейрон спрацьовує і видає на виході 1, що означає істину. В іншому випадку він залишається неактивним і видає 0, що означає неправду.
3. Булеві операції – логіку МР-нейронів можна зрозуміти з точки зору базових логічних операцій, таких як І, АБО та НЕ. Наприклад, у МР-нейроні з двома входами нейрон може спрацьовувати (видавати 1 або істину), тільки якщо обидва входи активні (вхід 1 І вхід 2). Або ж він може спрацьовувати, якщо активний будь-який з входів (вхід 1 АБО вхід 2). Крім того, поріг діє як форма заперечення, дозволяючи нейрону залишатися неактивним (виводячи 0, або брехню), коли сума входів не відповідає пороговій умові.

Застосування булевої логіки в моделі МР-нейрона дає основу для розуміння того, як прості обчислювальні блоки можуть виконувати логічні операції та приймати рішення на основі двійкових входів. Розглянемо формулу для виводу МР-нейрона:

$$Output = \begin{cases} 1, & \text{if } \sum_{i=1}^n x_i \geq threshold \\ 0, & \text{otherwise} \end{cases}$$

У цій формулі  $x_i$  позначає  $i$ -й вхід нейрона, а  $n$  – загальну кількість входів. Поріг – це параметр, який визначає рівень активації, необхідний для

того, щоб нейрон видав вивід. Якщо сума входів перевищує поріг, нейрон спрацьовує і видає на виході 1, інакше він залишається неактивним і видає 0.

Незважаючи на те, що МР-нейрон є фундаментальною концепцією в теорії нейронних мереж, він має кілька обмежень, які роблять його непрактичним для застосування в реальному світі. Одним з основних недоліків є його нездатність навчатися на основі даних. У моделі МР-нейрона поріг повинен бути встановлений вручну, що ускладнює адаптацію до різних завдань або наборів даних. Крім того, бінарна природа входів і виходів обмежує його здатність представляти складні взаємозв'язки в даних. Реальні проблеми часто вимагають більш нюансованого представлення та безперервних значень входів і виходів.

Щоб подолати обмеження МР-нейрона, Френк Розенблат створив перцептрон у 1958 році. Перцептрон є розширенням МР-нейрона, яке включає алгоритм навчання, що дозволяє йому автоматично налаштовувати свої ваги на основі спостережуваних даних. Цей алгоритм навчання дозволяє перцептрону вчитися на прикладах і робити прогнози на основі нових даних.

Перцептрон складається з входів, ваг, пов'язаних з кожним входом, зваженої суми входів і функції активації, яка визначає вихід. На відміну від МР-нейрона, перцептрон може навчатися оптимальним вагам, де він ітеративно підлаштовує ваги для мінімізації помилок прогнозування. Схему простого перцептрона зображено на рисунку 2.2.

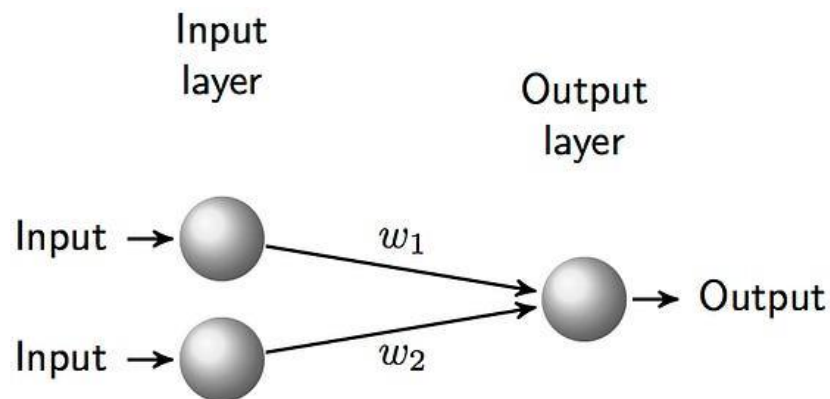


Рисунок 2.2 – Схема простого перцептрона [11]



Розробка перцептрона ознаменувала значний прогрес у теорії нейронних мереж, проклавши шлях до більш досконалих моделей, здатних вивчати складні закономірності на основі даних. Хоча перцептрон має свої обмеження, такі як нездатність вирішувати нелінійні задачі, він заклав основу для подальших досліджень у галузі штучних нейронних мереж і машинного навчання.

Підводячи підсумки, хоча МП-нейрон забезпечує фундаментальне розуміння нейронних обчислень, його обмеження роблять його непрактичним для застосування в реальному світі. Розробка перцептрона усунула деякі з цих обмежень, запровадивши алгоритм навчання, який дозволяє нейронним мережам адаптуватися до даних [12].

## 2.3 Способи оптимізації кількості кластерів

### 2.3.1 Метод ліктя

Кластеризація є неконтрольованою технікою машинного навчання, що передбачає розподіл набору даних на групи, в яких елементи однієї групи мають подібні характеристики. Серед найпоширеніших методів кластеризації можна виділити К-середні (K-Means), ієрархічну кластеризацію, кластеризацію на основі щільності та кластеризацію на основі моделей.

Метод ліктя використовується в аналізі даних і машинному навчанні для визначення оптимальної кількості кластерів у наборі даних. Цей метод передбачає побудову графіка, що відображає дисперсію в залежності від кількості кластерів, і виявлення точки «ліктя», де швидкість зменшення дисперсії різко сповільнюється. Це дозволяє визначити відповідну кількість кластерів для подальшого аналізу або навчання моделі.

Цей метод є візуальною технікою, яка використовується для визначення оптимального значення  $K$  в алгоритмі кластеризації методом  $k$ -середніх. У даному підході на графіку, відомому як ліктьовий графік, відображаються значення внутрішньокластерної суми квадратів (WCSS) в залежності від різних значень  $K$ . Оптимальне значення  $K$  визначається в точці, де графік має вигин, подібний до ліктя.

У методі Elbow кількість кластерів ( $K$ ) змінюється від 1 до 10. Для кожного значення  $K$  обчислюється WCSS (сума квадратів відстаней всередині кластеру). WCSS представляє собою суму квадратів відстаней між кожною точкою та центроїдом відповідного кластера. При побудові графіка WCSS в залежності від  $K$ , він набуває форми ліктя. При збільшенні значення  $K$  значення WCSS знижується. Найбільше значення WCSS спостерігається при  $K = 1$ . Аналізуючи графік на рисунку 2.3, можна помітити, що в певний момент графік різко змінюється, формуючи таким чином вигин, схожий на лікоть. З цієї точки графік рухається майже паралельно осі  $X$ . Значення  $K$ , що відповідає цій точці, є оптимальним значенням  $K$  або оптимальною кількістю кластерів [13].

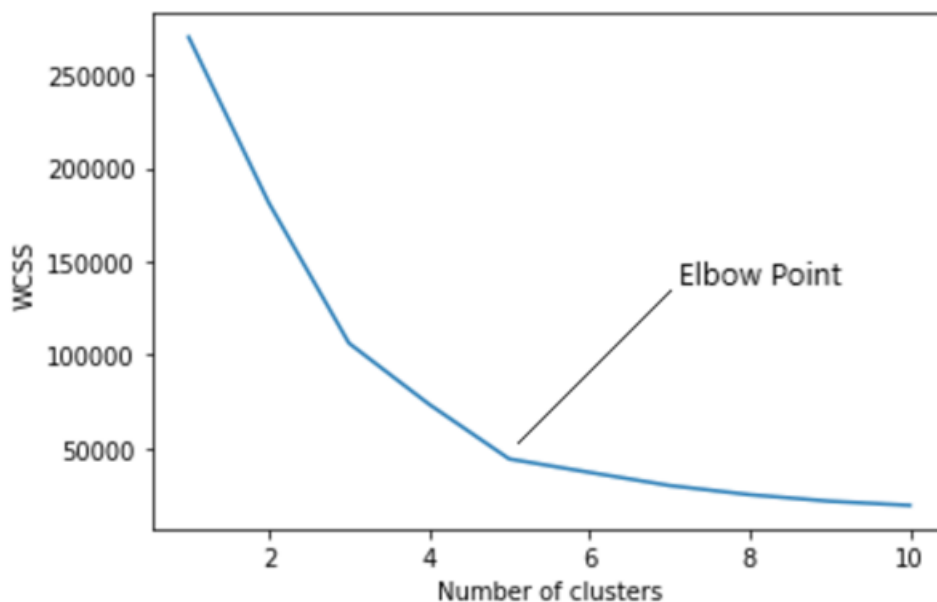


Рисунок 2.3 – Приклад графіку методу ліктя

### 2.3.2 Метод Silhouette score

Оцінка Silhouette є метрикою, що використовується для визначення якості результатів кластеризації даних. Цей показник розраховується шляхом вимірювання схожості кожної точки даних з кластером, до якого вона входить, а також її відмінності від інших кластерів. Оцінка Silhouette зазвичай застосовується для оцінки ефективності алгоритмів кластеризації, таких як K-means.

Процес обчислення оцінки Silhouette відбувається в декілька етапів. Спочатку для усіх точки даних вираховується середня відстань ( $a_i$ ) до інших точок у межах того ж кластера. Це значення відображає ступінь схожості даної точки з іншими точками в кластері. Далі для цих точок даних розраховується середня відстань ( $b_i$ ) до всіх інших кластерів, до яких вона не належить. Це значення показує, наскільки точка даних відрізняється від точок в інших кластерах. Шляхом обчислення середнього значення оцінки Silhouette, розрахованих для кожної точки даних, отримують загальну оцінку Silhouette, який вимірює успішність результатів кластеризації.

Оцінка Silhouette обчислюється за формулою:

$$\frac{(b_i - a_i)}{\max(a_i, b_i)}$$

Оцінки Silhouette варіюється від -1 до +1. Позитивні значення вказують на те, що точки даних належать до правильних кластерів, що свідчить про хороші результати кластеризації. Тоді як від'ємні значення вказують на те, що точки даних належать до неправильних кластерів, що свідчить про погані результати кластеризації. Нульове значення зазвичай свідчить про перекриття кластерів або про те, що точки даних однаково близькі до кількох кластерів.

Таким чином, оцінка Silhouette є важливим критерієм для оцінки налаштувань і результатів алгоритмів кластеризації даних. Високий показник Silhouette вказує на більш послідовні та якісні результати кластеризації, тоді як низький показник може свідчити про те, що точки даних віднесено до неправильних кластерів або що алгоритм кластеризації не підходить для даних [14].

## 2.4 Функції активації

Активаційні функції в нейронних мережах та глибокому навчанні призначені для активації прихованих вузлів що дозволяє отримати більш бажані результати. Основна мета функції активації полягає у введенні нелінійності в нейронну мережу, що дозволяє їй вивчати більш складні закономірності з вхідних даних.

### 2.4.1 Функція активації ReLU

Зрізаний лінійний вузол (англ. rectified linear unit, ReLU) – це кусково-лінійна функція, яка має властивість виводити вхідний сигнал без змін, якщо він позитивний, інакше виводиться нуль:

$$ReLU(x) = \max(0, x)$$

Для багатьох типів нейронних мереж вона стала функцією активації за замовчуванням, тому що при її використанні мережа легше навчається і часто видає кращу продуктивність.

Цю функцію активації вперше представлено в динамічній мережі Ганлозера у 2000 році, базуючись на біологічному та математичному обґрунтуванні. Вона була вперше продемонстрована в 2011 році як метод, що дозволяє ефективніше навчати глибші мережі в порівнянні з популярними функціями активації, такими як синусоїда і гіперболічний тангенс.

У 2017 рік ReLU став найпоширенішою функцією активації для глибоких нейронних мереж.

Основна причина, чому ReLU не використовували до недавнього часу, полягає в тому, що вона не була диференційовною в нульовій точці. Дослідники, як правило, використовували диференційовані функції активації, такі як синусоїда і гіперболічний тангенс. Однак на теперішній час вважається, що ReLU є однією з найкращих функцій активацій для глибокого навчання [15].

Графічно функція ReLU зображена на рисунку 2.4 [16].

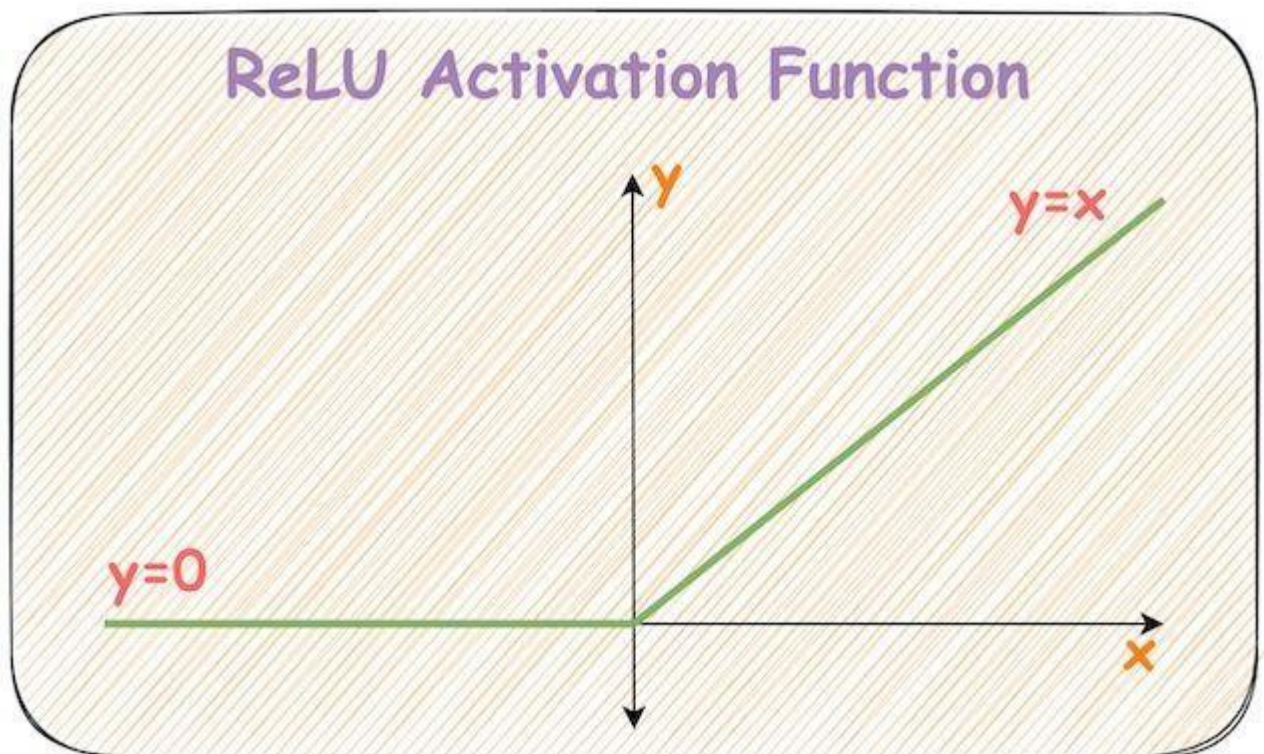


Рисунок 2.4 – Графічне представлення функції ReLU

## 2.4.2 Функція активації ELU

ELU – це функція активації, яка базується на ReLU, але має додаткову альфа-константу ( $\alpha$ ), що дозволяє отримувати результат при від'ємних вхідних даних. Графічно дана функція активації зображена на рисунку 2.5.

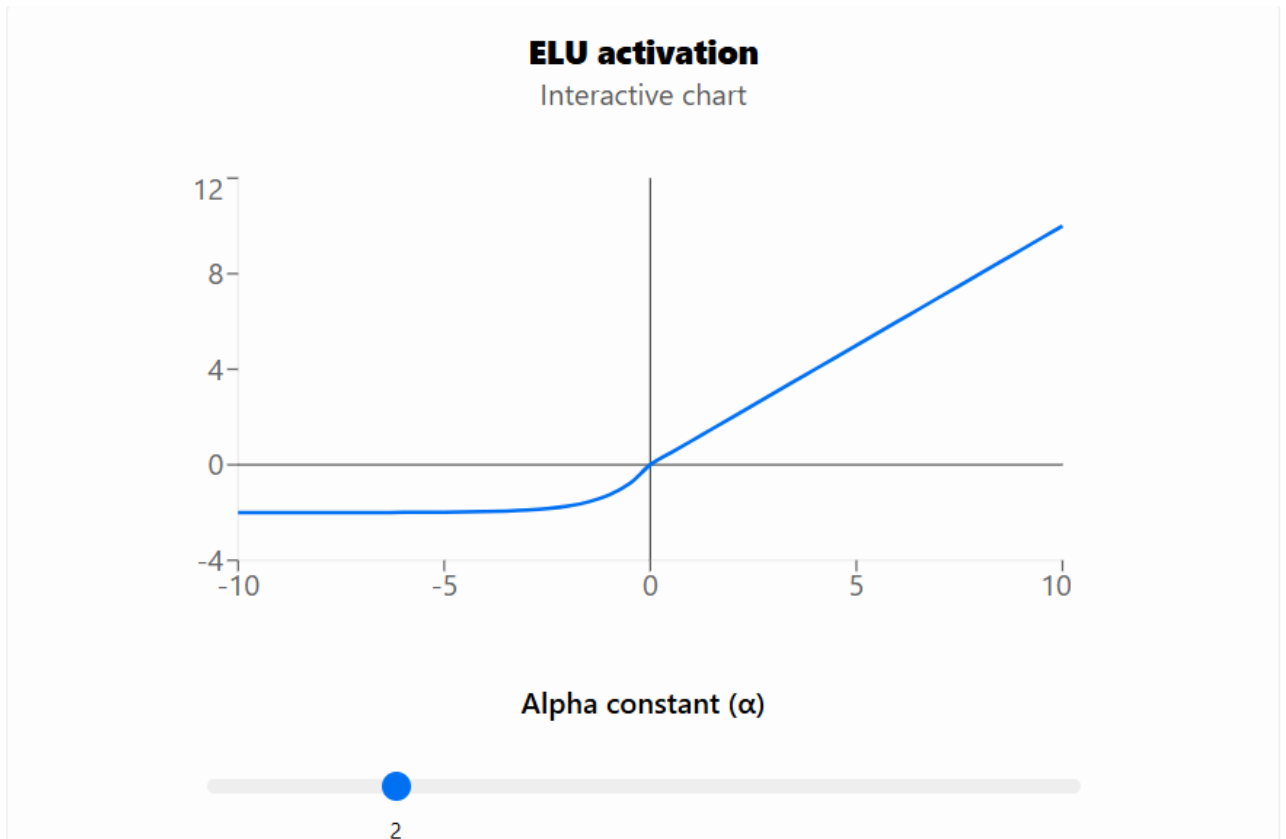


Рисунок 2.5 – Графік функції активації ELU

У вигляді формули функція активації ELU має такий вигляд:

$$ELU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha(e^x - 1), & \text{if } x < 0 \end{cases}$$

Для позитивних значень функція ELU виводить  $x$ . Тоді як при від'ємних вихідна крива злегка згладжується в напрямку до  $\alpha$ , де  $\alpha$  є константою. Чим вища альфа-константа, тим більш від'ємним стає вивід для від'ємних значень [17].

### 2.4.3 Функція активації Leaky ReLU

Негерметичний випрямлений лінійний блок (Leaky ReLU) – це функція активації, яка використовується в штучних нейронних мережах і є варіантом функції активації ReLU. Функція активації Leaky ReLU визначається наступним чином:

$$f(x) = \max(\alpha x, x)$$

де  $\alpha$  – невелика додатна константа (зазвичай встановлюється на рівні 0.01).

Коли функція Leaky ReLU приймає вхідні дані ( $x$ ) більші за 0, функція повертає саме вхідне значення, як і стандартна функція ReLU. Але коли вхідні дані ( $x$ ) менші або дорівнюють 0, замість того, щоб повертати 0 (як у випадку стандартної функції ReLU), Leaky ReLU повертає невелике від'ємне значення, пропорційне вхідним даним. Це визначається шляхом множення вхідного значення на малу константу  $\alpha$ .

Ключова відмінність Leaky ReLU від стандартного ReLU заключається в тому, що Leaky ReLU дозволяє невеликий градієнтний потік для від'ємних вхідних значень, в той час як ReLU встановлює вихід на 0 для від'ємних вхідних значень. За рахунок чого можна вирішити проблему «вмираючого ReLU», коли певні нейрони в мережі можуть назавжди вимкнутись і припинити навчатись під час навчання, якщо вони постійно отримують від'ємні вхідні дані.

Дозволяючи невеликий градієнтний потік для негативних входів, Leaky ReLU допомагає запобігти застряганню нейронів і дозволяє їм продовжувати навчання та оновлювати свої ваги в процесі навчання [18]. На рисунку 2.6 наведено порівняння графіків функції активації Leaky ReLU та ReLU [19].

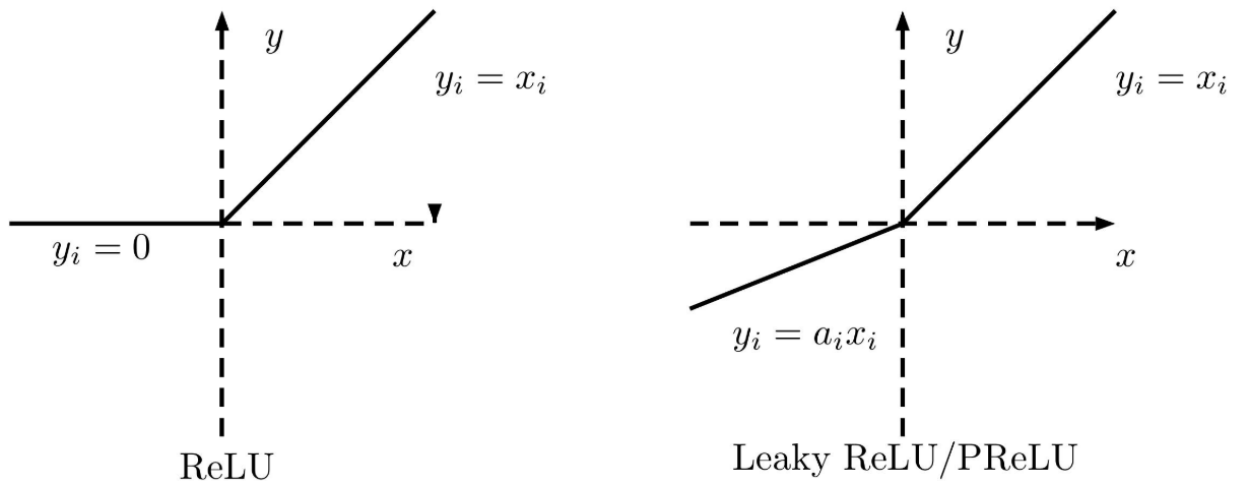


Рисунок 2.6 – Графіки функцій активації Leaky ReLU та ReLU

#### 2.4.4 Функція активації Softmax

Функція softmax, яка часто використовується в останньому шарі моделі нейронної мережі для задач класифікації, перетворює необроблені вихідні оцінки, також відомі як логічні значення, у ймовірності, беручи експоненту кожного вихідного значення і нормалізуючи ці значення шляхом ділення на суму всіх експонент. Цей процес гарантує, що вихідні значення знаходяться в діапазоні (0,1) і в сумі дорівнюють 1, що робить їх інтерпретованими як ймовірності.

Математичний вираз для функції softmax виглядає наступним чином:

$$\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Тут  $z_i$  представляє вхідні дані для функції softmax для класу  $i$ , а знаменник є сумою експоненціальних значень всіх необроблених оцінок класів у вихідному шарі.

При роботі функція активації softmax приймає на вхід вектор  $z$  дійсних чисел, що представляє виходи останнього шару нейронної мережі. Кожен



елемент  $u$  з підноситься до степеня з використанням математичної константи  $e$ . Це гарантує, що всі значення стануть додатними. Після чого піднесені до степеня значення діляться на суму всіх піднесених до степеня значень. Цей крок нормалізації гарантує, що сума вихідних значень дорівнюватиме 1, що є важливою властивістю розподілу ймовірностей [20]. Графік функції softmax можна побачити на рисунку 2.7 [21].

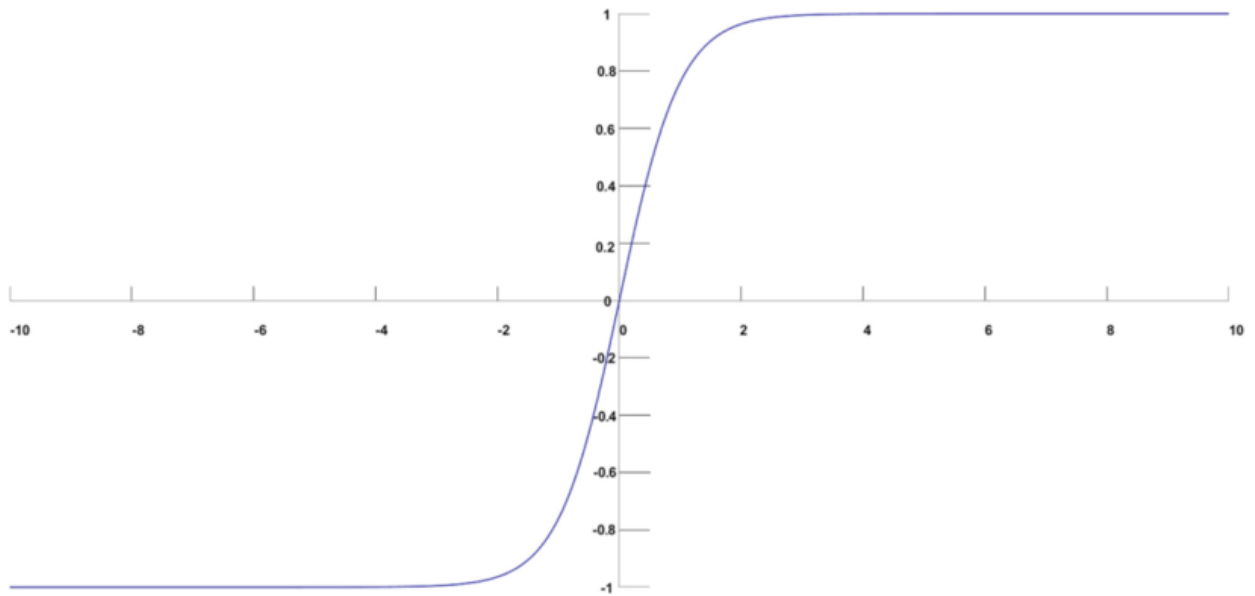


Рисунок 2.7 – Графік функції активації Softmax

## 2.5 Багатошарова нейронна мережа

Багатошарова нейронна мережа – це вдосконалена модель, яка використовується у штучному інтелекті та машинному навчанні. Дану мережу можна представити як набір взаємопов'язаних перцептронів. Однак, на відміну від перцептрона, який має лише один шар нейронів, багатошарова нейронна мережа має кілька шарів, розташованих один над одним. Кожен нейрон обчислює зважену суму своїх вхідних значень і пропускає її через функцію активації для отримання вихідного значення. Це дозволяє мережі вловлювати складні взаємозв'язки між входами та виходами. Усі шари можна умовно

розділити на три типи. Перший шар, який отримує вхідні дані називається вхідним шаром. Далі ідуть приховані шари, таких шарів у нейронній мережі може бути декілька. А також вихідних шар, останній шар, який видає вихідні дані мережі. Типову структуру багатошарової нейронної мережі можна побачити на рисунку 2.8.

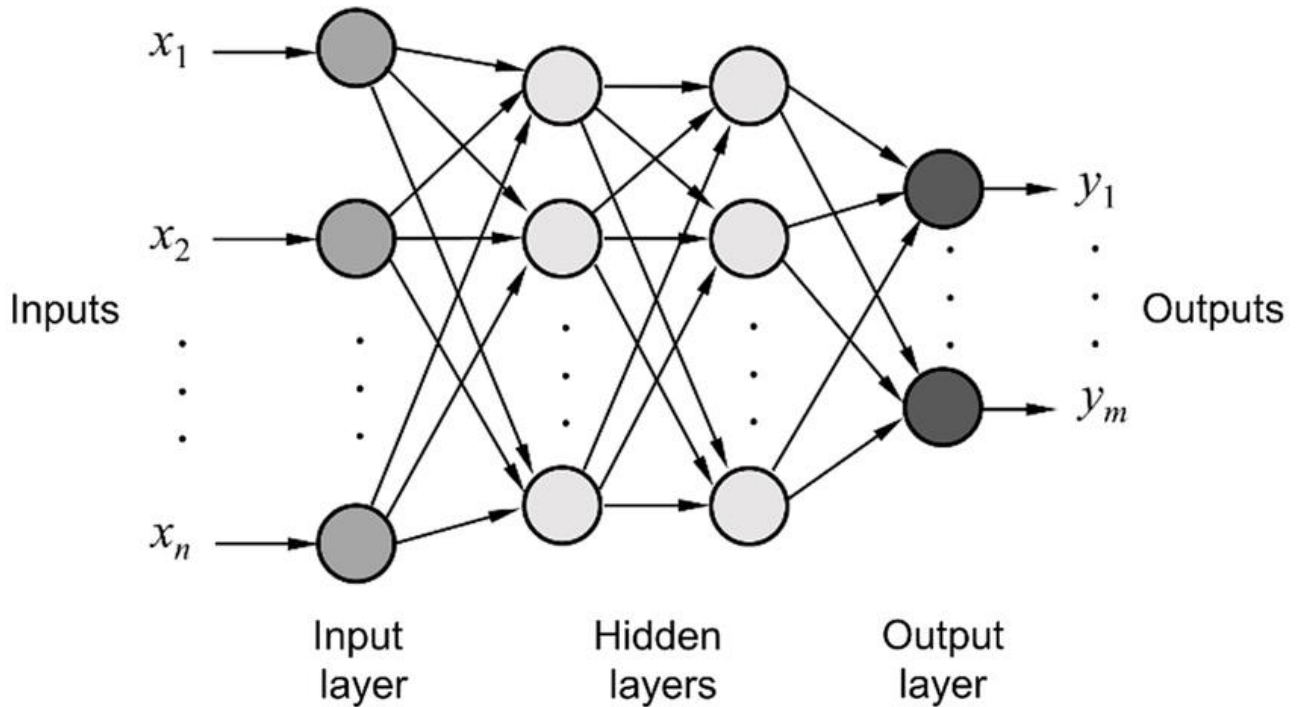


Рисунок 2.8 – Типова структура багатошарової нейронної мережі

На рисунку 2.8 зображується стандартна структура нейронної мережі прямого поширення, що складається з двох прихованих шарів. Мережа починається з вхідного шару, який отримує змінні  $x_1, x_2, \dots, x_n$ . Ці вхідні дані проходять через приховані шари, де відбувається навчання та налаштування мережі. Нарешті, вони досягають вихідного шару, даючи результати  $y_1, \dots, y_m$ . Кожен шар складається з нейронів, пов'язаних між собою, кожен з яких обчислює окремі операції, що дозволяє виявити складні закономірності та взаємозв'язки в даних [22].

Щоб змусити мережу навчатися, використовується зворотне поширення помилки – техніку, яка коригує ваги, що з'єднують нейрони, на основі похибки між прогнозованим і фактичним виходом. Це коригування

контролюється за допомогою градієнтного спуску або інших алгоритмів. Завдяки цьому ітеративному процесу мережа покращує свою здатність робити прогнози, що в кінцевому підсумку дозволяє їй вирішувати складні завдання [23].

## 2.6 Мережа довгої короткочасної пам'яті (LSTM)

Мережа довгої-короткочасної пам'яті (англ. long short-term memory, LSTM) – це тип рекурентних нейронних мереж, які можуть утримувати довгострокові залежності в послідовних даних. LSTM здатні обробляти та аналізувати послідовні дані, такі як текст, часові ряди і мова. Вони використовують комірку пам'яті та ворота для керування потоком інформації, що дозволяє їм вибірково зберігати або відкидати інформацію за потреби і, таким чином, уникати проблеми зникаючого градієнта, з якою стикаються традиційні RNN.

Рекурентні нейронні мережі (RNN) – це тип нейронних мереж, які призначені для обробки послідовних даних. Вони можуть аналізувати дані з часовим виміром, такі як часові ряди, мова і текст. RNN може робити це, використовуючи прихований стан, який передається від одного часового кроку до іншого. Прихований стан оновлюється на кожному часовому кроці на основі вхідних даних і попереднього прихованого стану. RNN здатні вловлювати короткострокові залежності в послідовних даних, але їм важко вловлювати довгострокові залежності [24].

Кожна модель сімейства RNN, включаючи LSTM, являє собою ланцюжок нейронів, що повторюються в своїй основі. У стандартних RNN кожен шар нейронів виконує лише одну операцію над вхідними даними. На рисунку 2.9 зображено схему стандартної RNN.

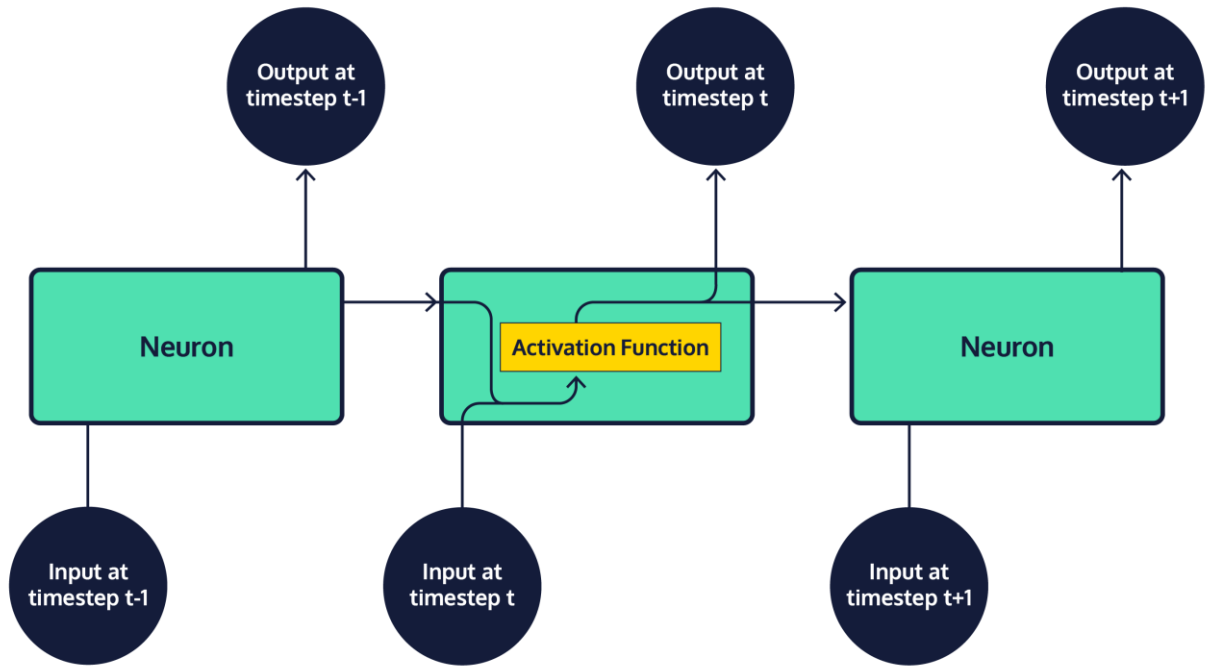


Рисунок 2.9 – Схема стандартної RNN

Однак у LSTM групи нейронів виконують чотири окремі операції над вхідними даними, які потім послідовно об'єднуються. На рисунку 2.10 зображена схема LSTM.

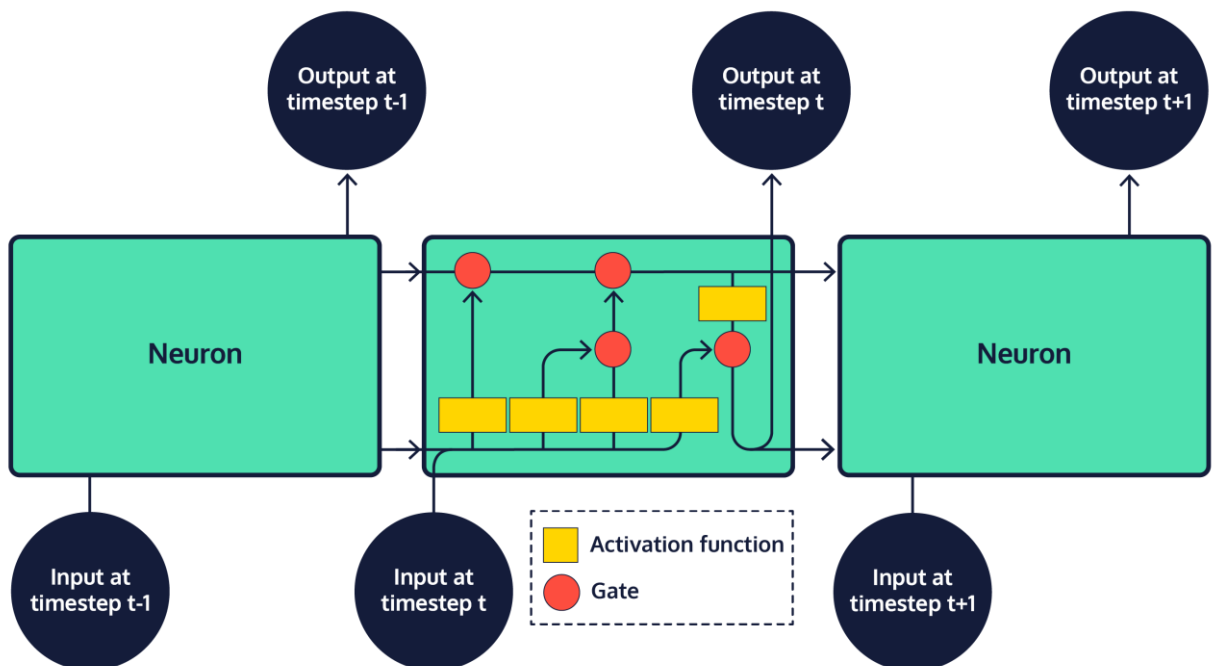


Рисунок 2.10 – Схема LSTM

Найважливішим аспектом LSTM є спосіб, у який перетворені вхідні дані комбінуються шляхом додавання результатів до пам'яті станів, або комірок, представлених у вигляді векторів. Існує два стани, які створюються на першому кроці послідовності, а потім переносяться при обробці наступних вхідних даних: стан комірки і прихований стан. На рисунку 2.11 зображена схема LSTM з виділеним станом комірки.

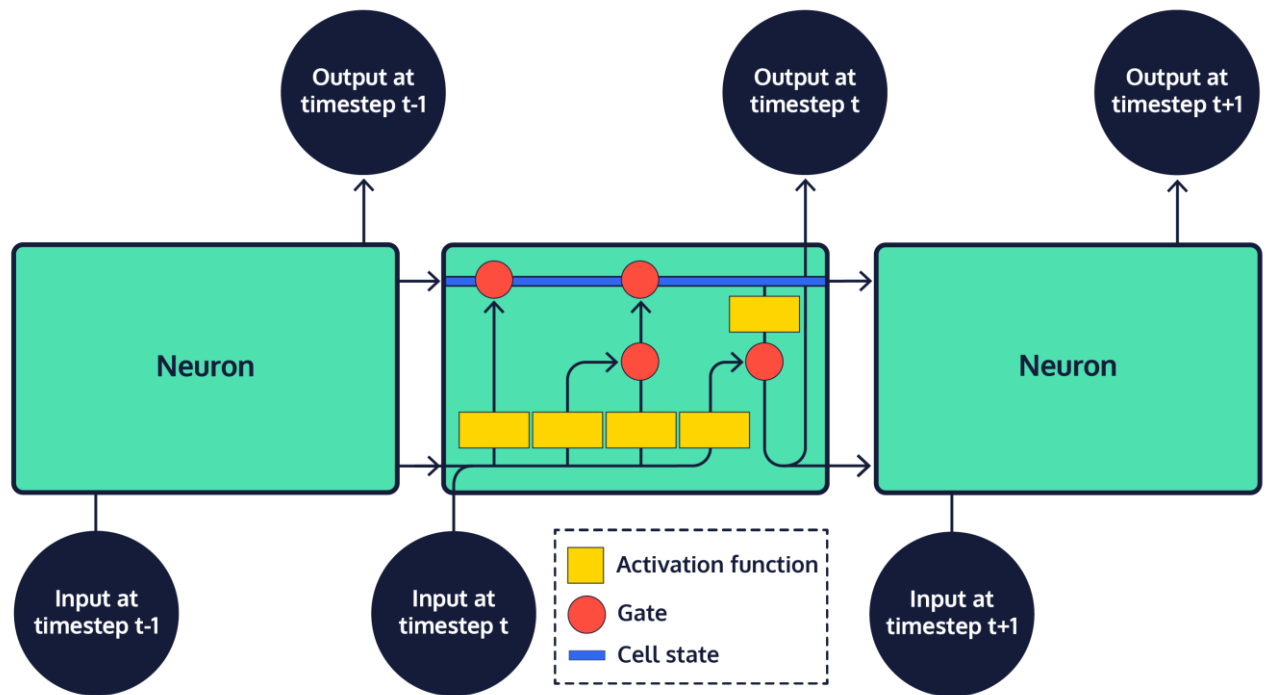


Рисунок 2.11 – Схема LSTM з виділеним станом комірки

Стан комірки переносить інформацію через мережу, коли ми обробляємо послідовність входів. На кожному часовому кроці, або кроці послідовності, оновлений вхід додається до стану комірки за допомогою воріт, які контролюють, яка частина входу повинна бути включена в кінцевий продукт стану комірки. Цей кінцевий продукт, який подається на вхід наступного шару нейронної мережі на наступному часовому кроці, називається прихованим станом. Кінцевий вихід нейронної мережі часто є результатом, що міститься в кінцевому прихованому стані, або середнім значенням результатів усіх прихованих станів мережі.

Збереження більшості станів комірки при перетворенні даних у поєднанні з інкрементними додаваннями, контрольованими воротами, дозволяє зберігати важливу інформацію з початкових вхідних даних у нейронній мережі. Зрештою, це дозволяє використовувати інформацію з більш ранніх вхідних даних для прийняття рішень у будь-якій точці моделі [25].

## 2.7 Random forest

Випадкові ліси (Random forests) або випадкові дерева рішень (Random Decision Trees) – це поєднання дерев рішень (decision trees), які працюють разом для отримання єдиного результату. Виникнувши в 2001 році завдяки Лео Брейману, Random Forest набув великою популярності серед ентузіастів машинного навчання.

Алгоритм випадкового лісу на етапі навчання створює ряд дерев рішень. Кожне дерево створюється на основі випадкової підмножини даних, а також, для кожного розділу використовується випадкова підмножина ознак. Завдяки цій випадковості забезпечується варіативність між окремими деревами, що знижує ймовірність перенавчання та підвищує загальну ефективність прогнозування.

При прогнозуванні алгоритм збирає результати всіх дерев або шляхом голосування (для задач класифікації), або шляхом усереднення (для задач регресії). Цей спільний процес прийняття рішень, підтриманий декількома деревами з їхніми знаннями, забезпечує стабільні результати. Випадкові ліси широко використовуються для класифікації та регресійних функцій, які відомі своєю здатністю обробляти складні дані, зменшувати перенавчання та надавати прогнози в різних середовищах. На рисунку 2.12 зображена схема алгоритму.

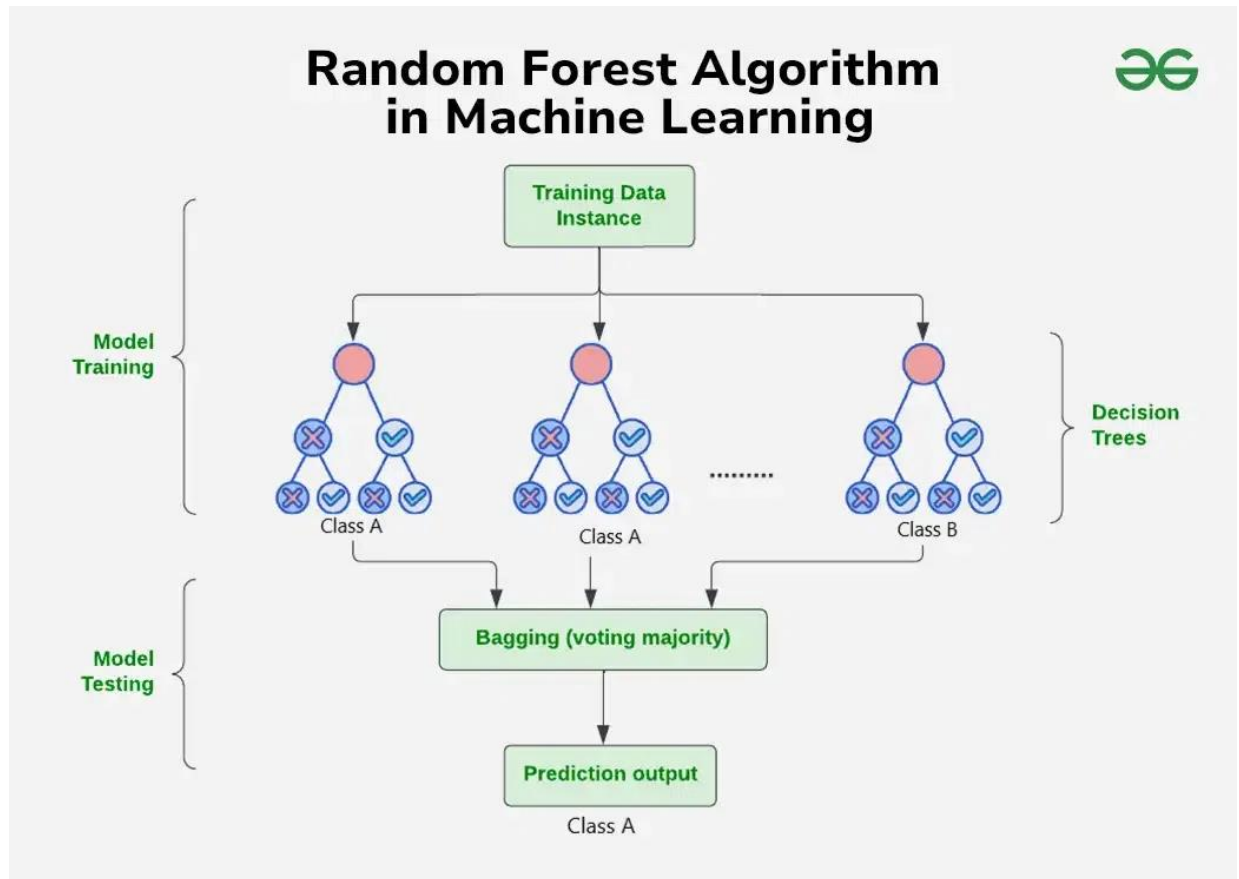


Рисунок 2.12 – Схема алгоритму Random Forest

Робота алгоритму випадкового лісу заключається в декількох етапах. Спочатку Random Forest використовує принципи ансамблевого навчання для формування групи дерев рішень. Ці дерева можна порівняти з окремими експертами, кожен з яких спеціалізується на певному аспекті даних. Важливо, що вони функціонують незалежно, що зменшує ризик надмірного впливу специфічних нюансів одного дерева на загальну модель. Щоб забезпечити унікальність кожного дерева в ансамблі, Random Forest застосовує випадковий вибір ознак. Під час навчання кожного дерева обирається випадкова підмножина ознак. Ця випадковість дозволяє кожному дереву зосередитися на різних аспектах даних, що сприяє формуванню різноманітного набору предикторів в ансамблі. Далі алгоритм використовує техніку пакування, яка є ключовим елементом стратегії навчання Random Forest. Вона передбачає створення кількох Bootstrap-зразків з початкового набору даних, що дозволяє вибирати приклади із заміною. Це дозволяє отримати різні підмножини даних

для кожного дерева рішень, що вносить варіативність у процес навчання і робить модель більш надійною. Коли настає етап прогнозування, кожне дерево рішень у випадковому лісі висловлює свою думку. У випадках класифікації остаточний прогноз визначається модою (найпоширенішим прогнозом) серед усіх дерев. У задачах регресії обчислюється середнє значення прогнозів окремих дерев. Цей механізм голосування забезпечує збалансований і колективний підхід до прийняття рішень [26].

## 2.8 Методи візуалізації вхідних даних

Свічковий графік – це тип фінансового графіка, який відображає рухи ціни активу за визначений проміжок часу. Як випливає з назви, він складається зі свічок, кожна з яких відображає однаковий часовий інтервал. Період свічки можна встановити практично будь-який, це може бути декілька секунд, або років.

Хоча свічкові графіки можуть бути використані для аналізу різних типів даних, їхнє найпоширеніше призначення – спростити аналіз фінансових ринків. При правильному застосуванні ці інструменти можуть допомогти трейдерам оцінити ймовірність різних результатів руху ціни. Вони є корисними, оскільки дозволяють трейдерам та інвесторам формувати власні висновки на основі ринкового аналізу.

Для створення кожної свічки потрібні такі цінові показники, схематичне зображення елемента свічкового графіку зображено на рисунку 2.13.

1. Відкриття — перша зафіксована торгова ціна активу за певний період.
2. Максимум — найвища зафіксована торгова ціна активу за певний період.



3. Мінімум — найнижча зафіксована торгова ціна активу за певний період.
4. Закриття — остання зафіксована торгова ціна активу за певний період.

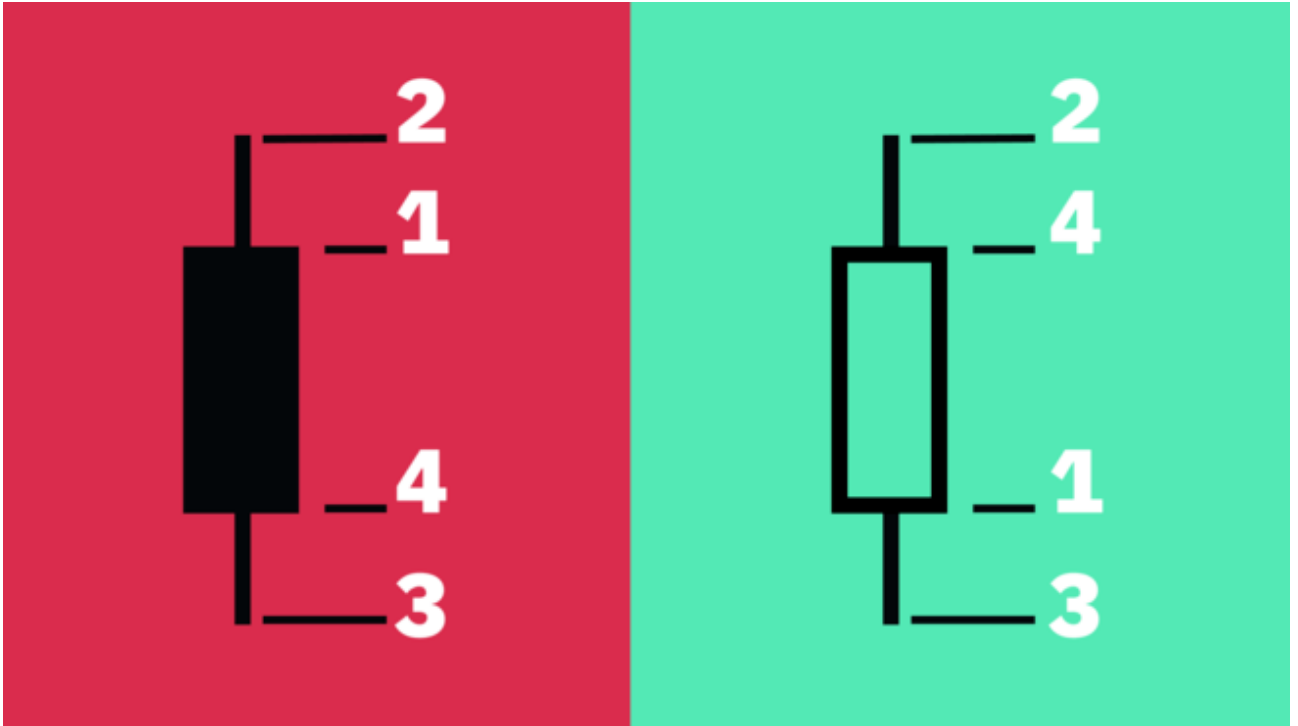


Рисунок 2.13 – Схематичне зображення елементу свічкового графіку

Зазвичай цю сукупність позначень називають OHLC. Співвідношення між цінами відкриття, максимуму, мінімуму та закриття формує тип свічки. Відстань між цінами відкриття та закриття називається тілом свічки, а відстань між тілом та максимумом або мінімумом – тінню. Відстань між максимумом і мінімумом свічки визначається як діапазон свічки [27].

Індикаторний аналіз – це метод, що дозволяє прогнозувати зміни цін, використовуючи певні індикатори для прийняття рішень.

Індикатори – це програми, які аналізують і інтерпретують цінову поведінку за допомогою математичних формул.

Скористаємось середній ковзним (Moving Average), який є одним з найпоширеніших індикаторів серед трейдерів на сьогодні.

Середня ковзна (Moving Average, MA) — це індикатор, що згладжує графік цін та вказує на напрямок основного тренду [28].

Відсоткова зміна цін на акції – це показник, який використовується для вираження зміни ціни з плином часу у відсотках до попередньої ціни. Цей показник корисний для порівняння ефективності акцій за різні періоди часу або порівняно з іншими акціями. Формула для розрахунку щоденної зміни у відсотках наступна:

$$\text{Percentage Change} = \left( \frac{\text{Current Price} - \text{Previous Price}}{\text{Previous Price}} \right) \times 100$$

де, *Current Price* – це ціна акції на кінець поточного періоду;

*Previous Price* – це ціна акції на кінець попереднього періоду.

Для першого періоду в наборі даних часового ряду (наприклад, перший день наявних біржових даних) відсоткова зміна не визначається, оскільки немає попередньої ціни для порівняння. У таких випадках зазвичай встановлюють нульове значення зміни у відсотках або не вказують це значення. При аналізі даних про запаси за більш тривалий період, наприклад, місяць або рік, відсоткова зміна розраховується з використанням ціни на початок і кінець періоду. Цей показник зазвичай використовується у фінансовому аналізі для оцінки волативності та ефективності акцій. Це також ключовий показник для інвесторів, які приймають рішення про купівлю або продаж цінних паперів.

## 2.9 Схема підходу

Робота реалізованого методу відбувається у декілька етапів, розглянемо схеми які зображені на рисунку 2.14 та 2.15.



Рисунок 2.14 – Схема підходу

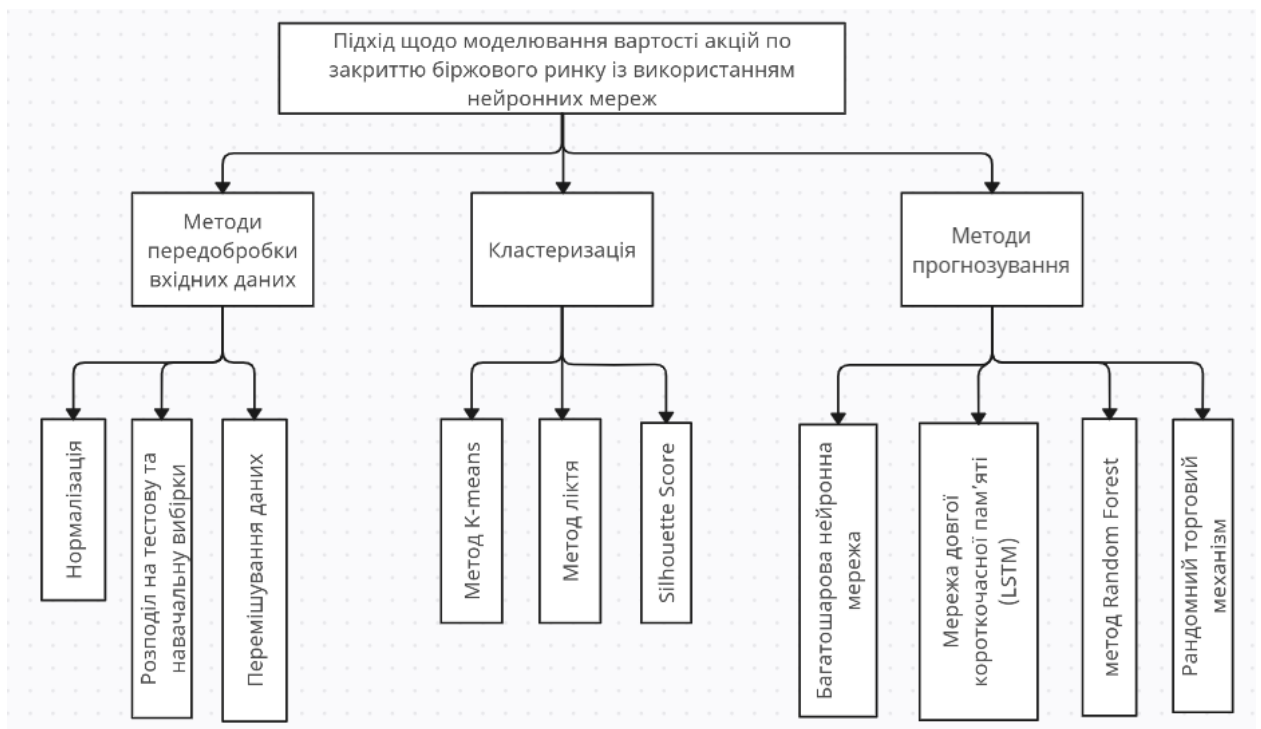


Рисунок 2.15 – Ієрархічна схема підходу

Перший кроком буде завантаження вхідних даних про стан акцій. Для даної роботи сайт «Yahoo! Finance» є важливий тим, що надає можливість завантажити датасет із даними про ціни акцій різних компаній, зокрема ціна

на початку та кінці календарного дня, найвища і найнижча ціна протягом дня, а також обсяг проданих акцій. Для аналізу та прогнозування була використана ціна акцій наприкінці кожного дня.

Другим кроком у прогнозуванні ціни акцій є передобробка вхідних даних. На даному етапі відбувається очистка вхідних даних від пропусків, після чого дані нормалізуються за допомогою стандартної нормалізації. Далі використовується кластеризація для сегментації даних на групи з подібними характеристиками волативності. Спочатку визначається оптимальна кількість кластерів, для цього використовуються два підходи.

1. Метод ліктя (Elbow): в даному методі обчислюється SSE (сума квадратів відстаней від кожної точки до центроїди її кластера). Залежність між кількістю кластерів і SSE відображає вигин на графіку (точку «ліктя»), який вказує на оптимальну кількість кластерів.
2. Silhouette Score: в даному методі оцінюється якість кластеризації шляхом порівняння відстаней між точками одного кластеру та іншими кластерами. Чим ближче значення до 1, тим якісніша кластеризація.

Результати обчислень аналізуються програмою та обирається кількість кластерів з найкращими результатами. Алгоритм K-Means групує всі точки даних у відповідні кластери, що дозволяє сегментувати фінансові дані за характеристиками їхньої волативності.

Після кластеризації дані кожного кластера розділяються на навчальні та тестові вибірки у співвідношенні 80% для навчання і 20% для тестування. Щоб забезпечити випадковість, перед розділенням дані в межах кожного кластера перемішуються для запобігання зміщення моделі, викликаного нерівномірним розподілом даних.

Далі відбувається навчання багатошарової нейронної мережі, мережі довгої-короткочасної пам'яті, методу Random Forest та рандомного торгового механізму. Протягом навчання мереж зберігаються найкращі ваги моделі, це

дозволяє уникнути погіршення моделі через перенавчання або шум. Після завершення навчання для кожного кластера виконується прогнозування на тестовій вибірці, а результати використовуються для оцінки продуктивності моделі.

Для оцінок якості роботи мереж на кожному кластері використано кілька ключових метрик, таких як:

1. True Positive Rate (TPR):

$$TPR = \frac{\text{True Positive (TP)} + \text{False Negative (FN)}}{\text{True Positive (TP)}}$$

2. False Positive Rate (FPR):

$$FPR = \frac{\text{False Positive (FP)} + \text{True Negative (TN)}}{\text{False Positive (FP)}}$$

3. True Negative Rate (TNR):

$$TNR = \frac{\text{True Negative (TN)} + \text{False Positive (FP)}}{\text{True Negative (TN)}}$$

4. False Negative Rate (FNR):

$$FNR = \frac{\text{True Positive (TP)} + \text{False Negative (FN)}}{\text{False Negative (FN)}}$$

5. RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

6. Precision:

$$Precision = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}$$

7. Recall:

$$Recall = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}}$$

8. F1-score:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

## 2.10 Висновки до розділу 2

У цьому розділі було розглянуто основи нейронних мереж та їх застосування для прогнозування цін акцій. Розглянуто такі методи кластеризації як метод ліктя та Silhouette score. Також розглянуто використані методи візуалізації вхідних даних та наведено схему реалізованого методу.

На основі розглянутих методів створено підхід, що складається з таких етапів: завантаження вхідних даних, передобробка вхідних даних, кластеризація, розподіл даних, навчання моделей та методів, прогнозування та оцінка продуктивності.

При побудові нейронних мереж важливо правильно обрати функції активації, тому було розглянуто такі широко використовувані функції, як ReLU, ELU, Leaky ReLU та Softmax. У кожній з цих функцій є свої плюси та мінуси в залежності від конкретного завдання.

Також були розглянуті багат шарові нейронні мережі, мережі довгої короткочасної пам'яті та алгоритм Random Forest. Описано принцип їх побудови та роботи.

Ці знання будуть використані при побудові моделей для передбачення вартості акцій по закриттю біржового ринку.

## РОЗДІЛ 3 ОГЛЯД ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

### 3.1 Вступ до розділу 3

У даному розділі буде представлена програмна реалізація багатовартової нейронної мережі, мережі довгої короткочасної пам'яті та методу Random Forest. Для реалізації було обрано мову програмування Python (версія 3.9.12) [29] та середу розробки Jupyter Notebook [30]. Дана комбінація була обрана, опираючись на великий набір бібліотек та фреймворків присутній у мові програмування Python, а також простоту роботи та візуалізації присутній в Jupyter Notebook. При роботі з нейронними мережами використовувались бібліотеки Numpy, Pandas, TensorFlow, Sklearn та Matplotlib. Також для отримання історичних даних про акції було використано бібліотеку уfinance. Дана бібліотека дозволяє брати інформацію про ціни на акції компаній з інтернет-ресурсу «Yahoo! Finance» [31].

### 3.2 Візуалізація вхідних даних

Приклад датасету отриманого з-за допомогою веб-сайту «Yahoo! Finance» можна розглянути на рисунку 3.1. Ціни акцій наведені у доларах США.

2024-09-13 00:00:00-04:00	156.362	159.275	156.110	158.370
2024-09-16 00:00:00-04:00	158.330	159.240	157.610	158.990
2024-09-17 00:00:00-04:00	160.090	161.590	159.410	160.280
2024-09-18 00:00:00-04:00	160.850	161.630	159.660	160.810
2024-09-19 00:00:00-04:00	164.820	164.990	162.522	163.240
2024-09-20 00:00:00-04:00	164.520	164.750	163.180	164.640
2024-09-23 00:00:00-04:00	165.340	166.610	162.950	163.070
2024-09-24 00:00:00-04:00	164.250	164.550	162.030	163.640
2024-09-25 00:00:00-04:00	162.970	164.217	162.775	162.990
2024-09-26 00:00:00-04:00	165.030	165.500	163.500	163.830
2024-09-27 00:00:00-04:00	163.910	166.970	163.830	165.290
2024-09-30 00:00:00-04:00	164.780	167.360	164.640	167.190
2024-10-01 00:00:00-04:00	168.860	170.440	165.900	168.420
2024-10-02 00:00:00-04:00	167.760	168.880	166.250	167.310
2024-10-03 00:00:00-04:00	165.820	167.910	165.370	167.210
2024-10-04 00:00:00-04:00	169.340	169.550	166.960	168.560
2024-10-07 00:00:00-04:00	169.140	169.900	164.130	164.390
2024-10-08 00:00:00-04:00	165.430	166.100	164.310	165.700
2024-10-09 00:00:00-04:00	164.855	166.260	161.120	163.060
2024-10-10 00:00:00-04:00	162.110	164.311	161.640	163.180
2024-10-11 00:00:00-04:00	163.330	165.270	162.500	164.520
2024-10-14 00:00:00-04:00	164.910	167.620	164.780	166.350

Рисунок 3.1 — Приклад частини датасету, що містить дані про акції

Для кращого розуміння графіку зміни ціни акцій скористаємось свічковим графіком наведеним на рисунку 3.2.





Рисунок 3.2 – Свічковий графік для акцій GOOG

Для подальшої обробки вхідних даних варто також провести індикаторний аналіз, графік зображено на рисунку 3.3.

Розглянемо 20, 50 і 200 МА. Розгляд різних періодів МА може допомогти з'ясувати куди рухається тренд. Наприклад, доцільно додати 200-денну та 50-денну середні ковзні. Ці числа вказують на періоди, за які обчислюється показник. Якщо 200-денна і 50-денна середні ковзні перетинаються, це може свідчити про можливий розворот тренду.



Рисунок 3.3 – Свічковий графік із індикатором «середня ковзна (Moving Average)»

Розглянемо також на рисунку 3.4 графік Stock Split (поділ акцій) який покаже скільки процесів поділу акцій відбулось за період існування компанії на ринку акцій.

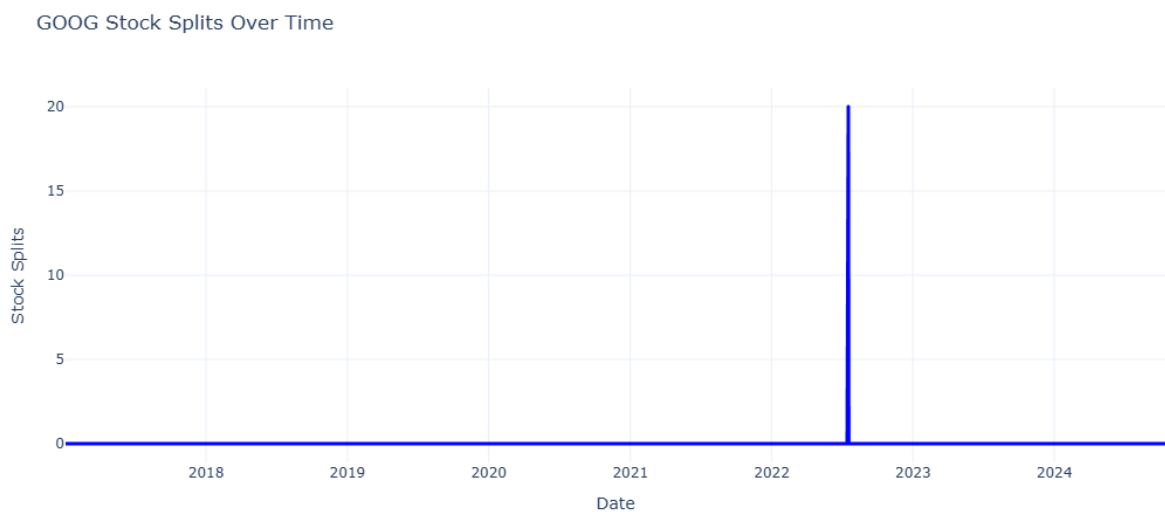


Рисунок 3.4 – графік Stock Split

Stock split (поділ акцій) — це процес, при якому компанія збільшує кількість своїх акцій, розділяючи кожен існуючу акцію на кілька нових. Наприклад, при розподілі 2:1 кожна акція стає двома, а їхня вартість знижується вдвічі. Це робиться для зниження ціни акцій, що робить їх більш доступними для інвесторів [32].

### 3.3 Відсоткова зміна цін на акції

Згідно рисунку 3.5 проведемо аналіз відсоткової зміни цін на акції.

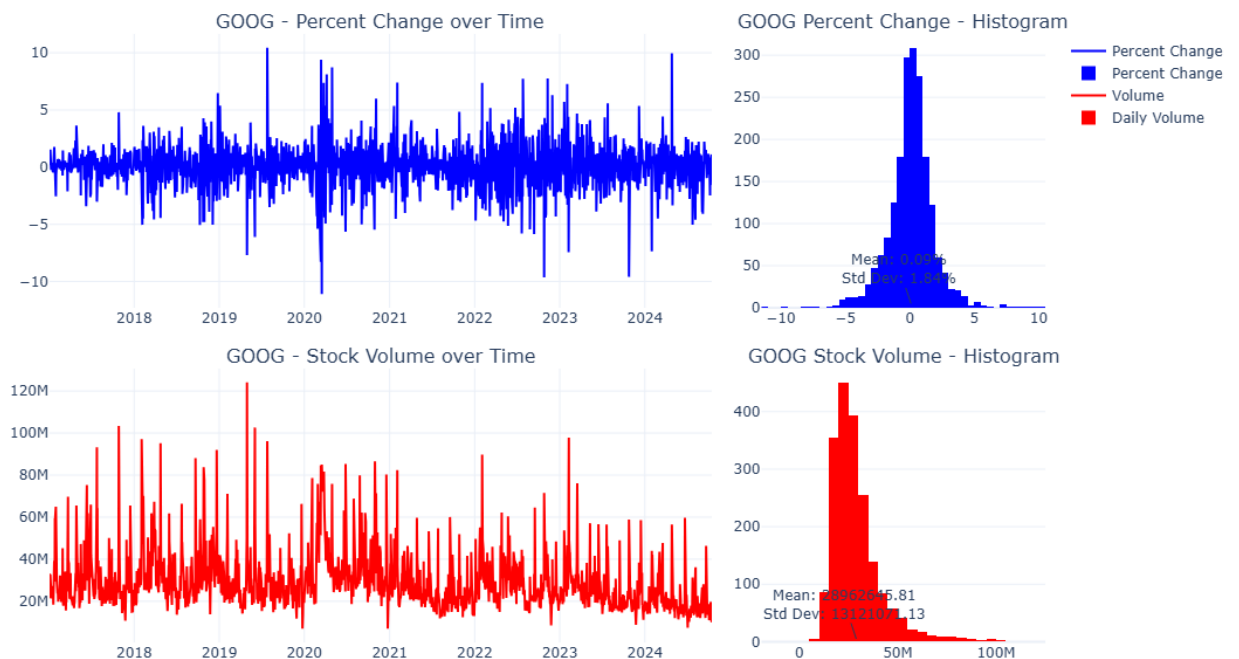


Рисунок 3.5 – Відсоткова зміна цін на акції GOOG

На рисунку 3.5 зображено.

1. GOOG – Percent Change over Time: графік показує щоденну відсоткову зміну ціни акцій Alphabet, показуючи волативність протягом спостережуваного періоду. Коливання відображаються

різкими сплесками і падіннями, що вказують на дні зі значними ціновими змінами. Це може відображати реакцію ринку на новинні події, звіти про прибутки або ширші економічні умови.

2. GOOG Percent Change – Histogram: графік у верхньому правому куті показує розподіл цих щоденних змін у відсотках. Більшість змін групуються навколо середнього значення, що свідчить про нормальний розподіл прибутковості, який є типовим для цін на акції з плином часу. Середнє значення, близьке до нуля, означає стабільне середнє зростання, тоді як стандартне відхилення вказує на ступінь відхилення від середнього значення.
3. GOOG – Stock Volume over Time: показує обсяг торгів акціями Alphabet. Піки на цьому графіку можуть відповідати певним подіям або виходу важливих новин, що впливають на настрої інвесторів і торговельну поведінку.
4. GOOG Stock Volume – Histogram: ілюструє розподіл обсягу торгів, показуючи, як часто трапляються певні обсяги. Середнє значення і стандартне відхилення дають уявлення про типовий обсяг і його мінливість. Концентрація даних у нижній частині графіка свідчить про те, що дні з високими обсягами трапляються рідше, але можуть бути пов'язані з ключовими подіями на ринку або в компанії.

Також проведемо аналіз оптимального числа кластерів з-за допомогою метода ліктя та Silhouette Score. Графічно результат обчислень можна побачити на рисунку 3.6.

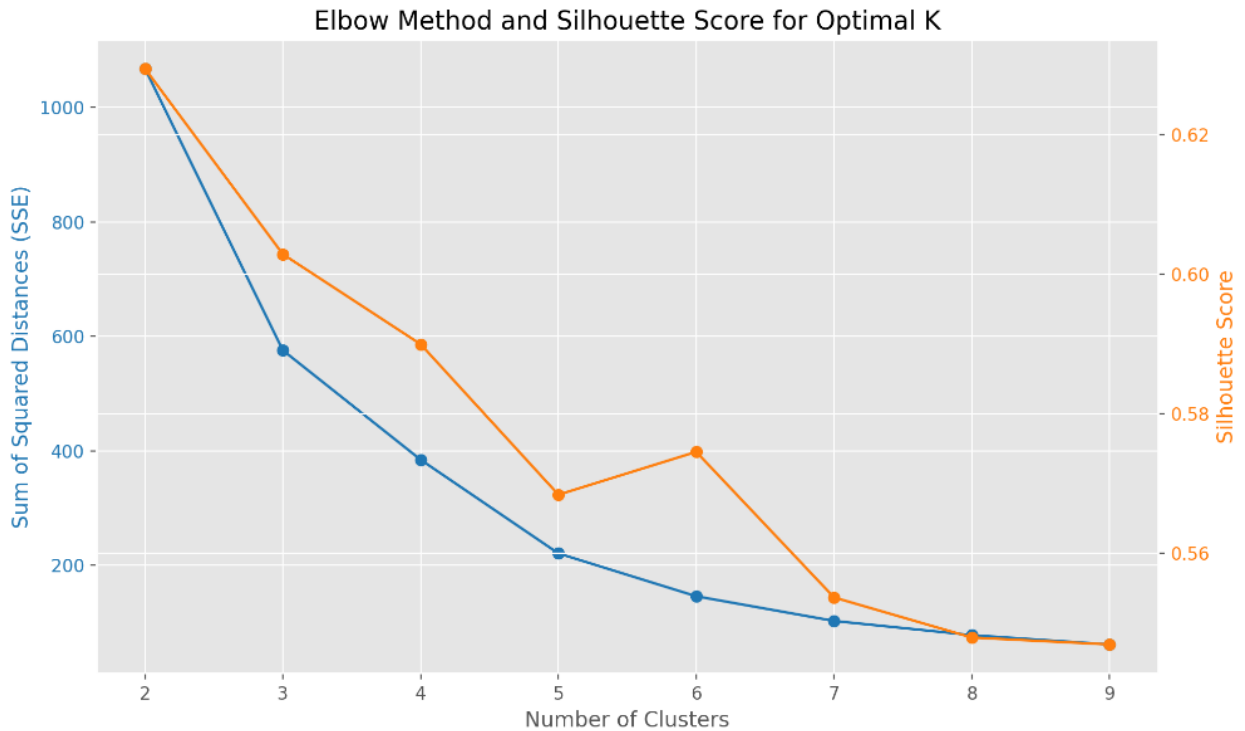


Рисунок 3.6 – Графіки підходів для визначення кількості кластерів

Розглядаючи графік 3.6 можна дійти висновку, що оптимальним числом кластерів для вхідного датасету буде 2. Оскільки, згідно графіку, Silhouette Score при двох кластерах має найбільше значення.

### 3.4 Реалізація багатошарової нейронної мережі

Для реалізації багатошарової нейронної мережі було вирішено обрати схему з трьома шарами.

Першим шар (вхідним) було обрано повнозв'язний шар з 64 нейронами. Для зменшення перенавчання використовується L2 регуляризація та Dropout із шансом відключити 40% нейронів. Після неї використовується BatchNormalization, щоб нормалізувати вхідні дані для цього шару. Як функцію активації вирішено використати ELU, щоб уникнути проблеми мертвих нейронів.

Другий шар (прихований) також являється повнозв'язним шаром з 32 нейронами. Для зменшення перенавчання використовується L2 регуляризація. Після неї використовується BatchNormalization і функція активації ELU.

Третій шар (вихідний) має два нейрони оскільки відбувається двокласова класифікація. Для активації цього шару використовується функція активації Softmax. Навчання моделі відбувається протягом 150 епох з використанням пакетного розміру 128.

Проведемо графічний аналіз отриманих результатів для двох кластерів зображених на рисунку 3.7 та 3.8.

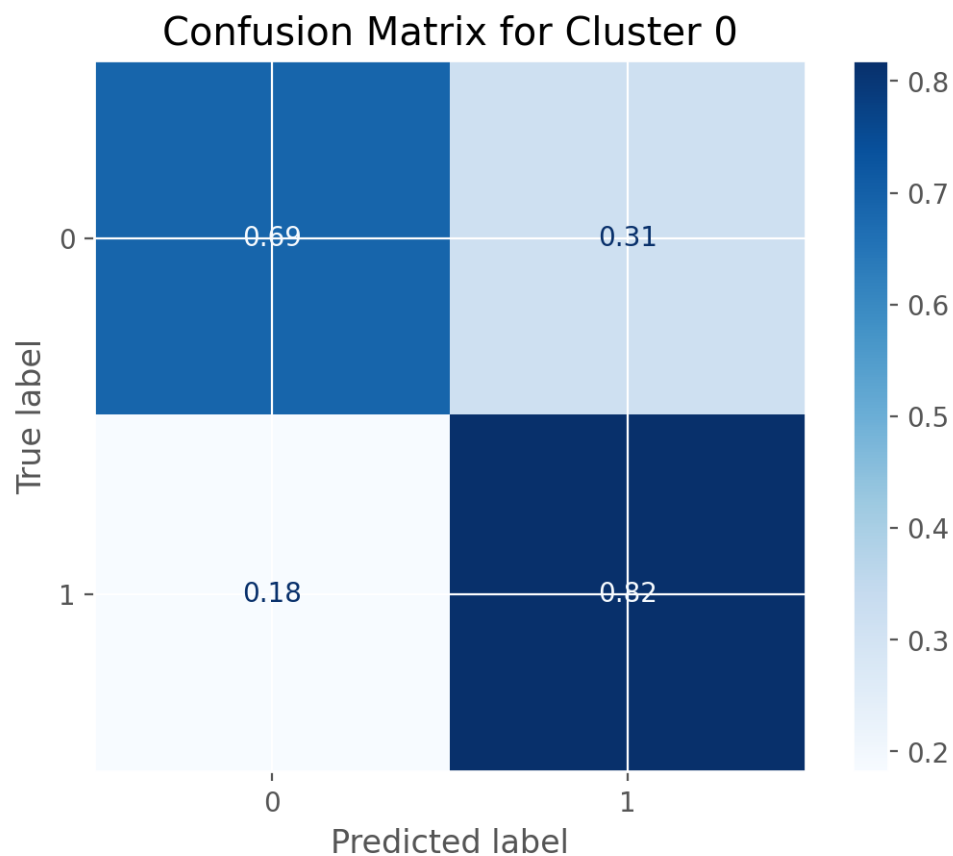


Рисунок 3.7 – Результати для кластеру 0

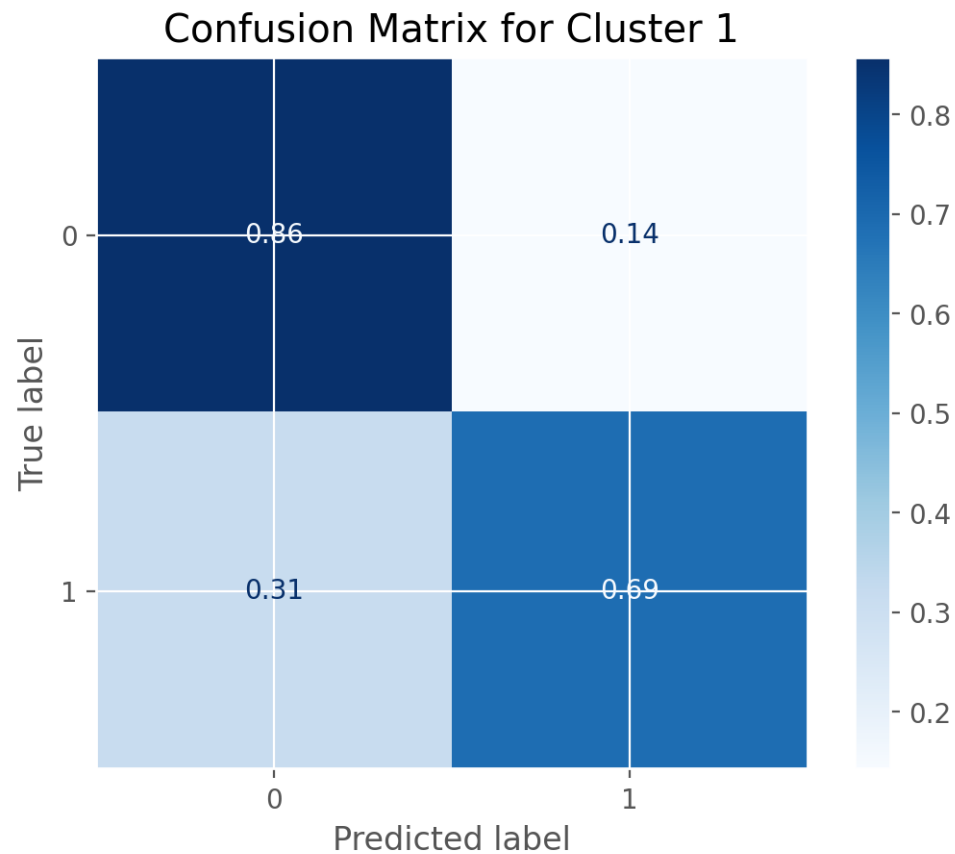


Рисунок 3.8 – Результат для кластеру 1

Згідно матриці похибок ми можемо побачити, що кількість неправдивих передбачень втричі нижча за кількість правдивих, що свідчить про достатньо якісну роботу мережі.

Тепер розглянемо метрики для результатів роботи багатосарової нейронної мережі які зображені на рисунку 3.9, 3.10, 3.11.

```
Cluster 0 - Classification Report:
              precision    recall  f1-score   support

     0       0.79         0.69         0.74         211
     1       0.72         0.82         0.77         209

 accuracy          0.75         420
 macro avg         0.76         0.75         0.75         420
 weighted avg      0.76         0.75         0.75         420

Cluster 0 - Confusion Matrix (normalized):
[[0.68720379 0.31279621]
 [0.18181818 0.81818182]]
Cluster 0 - RMSE: 0.4976133515281193
```

Рисунок 3.9 – Метрики для cluster 0

```

Cluster 1 - Classification Report:
      precision    recall  f1-score   support

     0       0.70      0.86      0.77        91
     1       0.85      0.69      0.76       106

 accuracy          0.77        197
 macro avg          0.78        197
 weighted avg       0.78        197

Cluster 1 - Confusion Matrix (normalized):
[[0.85714286 0.14285714]
 [0.31132075 0.68867925]]
Cluster 1 - RMSE: 0.48322100334222434

```

Рисунок 3.10 – Метрики для cluster 1

```

Summary of results for all clusters:
Cluster 0:
True positive rate = 0.687
False positive rate = 0.313
True negative rate = 0.818
False negative rate = 0.182
RMSE: 0.4976133515281193

Cluster 1:
True positive rate = 0.857
False positive rate = 0.143
True negative rate = 0.689
False negative rate = 0.311
RMSE: 0.48322100334222434

```

Рисунок 3.11 – Метрики для багатосарової нейронної мережі

Виходячи з отриманих результатів, можна зробити висновок, що модель cluster 0 має точність у 81% при передбаченні негативних класів, тоді як модель cluster 1 має точність 85% при передбаченні позитивних класів. Для обох моделей значення RMSE менше 0.5, що свідчить про невисокий рівень помилок.



### 3.5 Реалізація мережі довгої короткочасної пам'яті

Реалізована мережа довгої короткочасної пам'яті включає в себе основний LSTM шар, який включає 32 нейрони. Використовує L2 регуляризацію та Dropout щоб зменшити перенавчання.

Також є два додаткових шари, один з них це повнов'язний шар з 64 нейронами для подальшої обробки вихідних даних з LSTM. Функцією активації являється LeakyReLU, а також використовується BatchNormalization для нормалізації.

Останнім шаром являється шар з 2 нейронами для класифікації в два класи. Softmax використовується як функція активації.

Для прискорення навчання використовується оптимізатор Nadam. Реалізоване зниження швидкості навчання (Learning Rate Scheduler), що дозволяє автоматично знижувати швидкість навчання, якщо точність на валідаційних даних не покращується протягом 25 епох. За допомогою ModelCheckpoint зберігаються найкращі ваги моделі, коли модель досягає кращої точності на валідаційних даних, що дозволяє зменшити перенавчання.

Розглянемо результати роботи даної мережі на рисунках 3.12, 3.13, 3.14, 3.15 та 3.16.

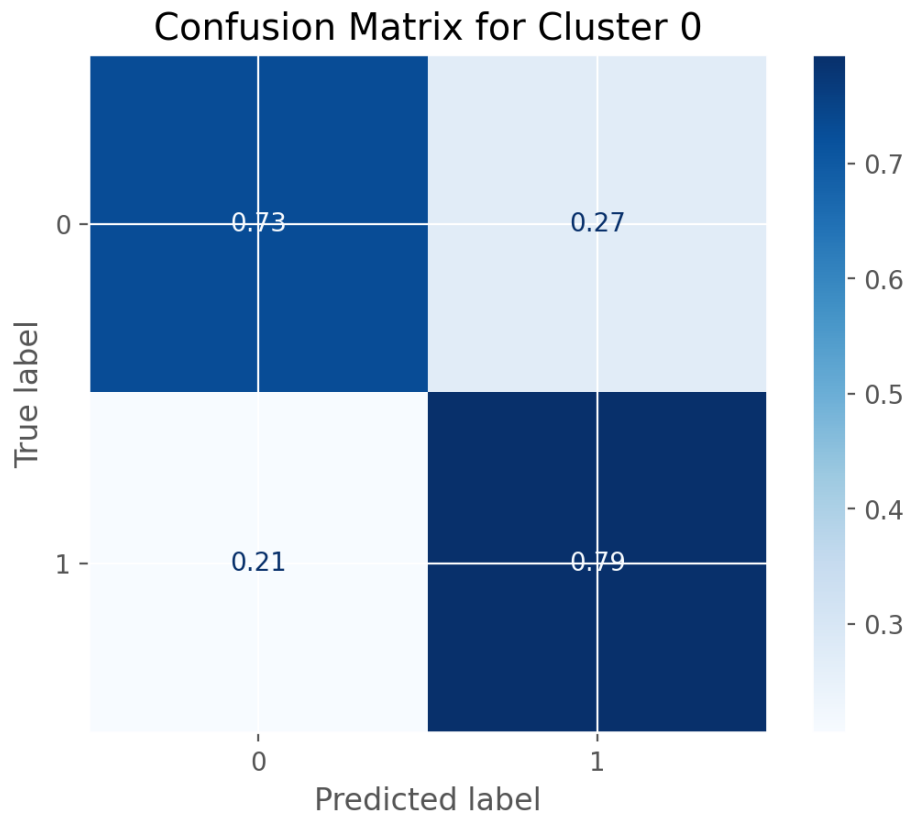


Рисунок 3.12 – Результат для кластеру 0

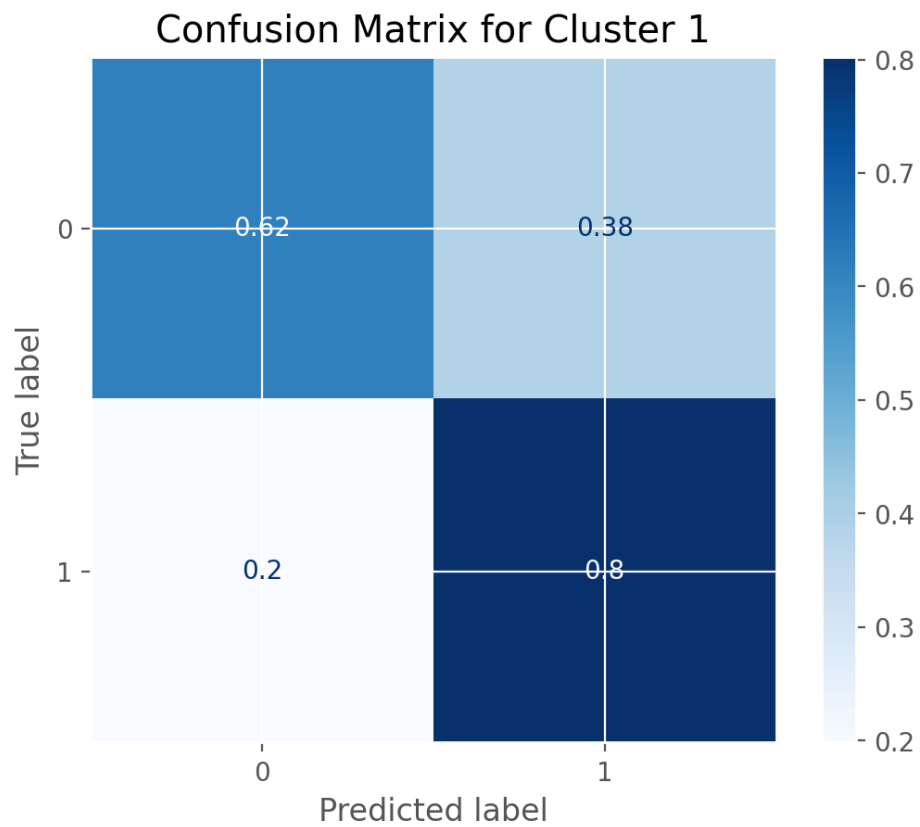


Рисунок 3.13 – Результат для кластеру 1

```

Cluster 0 - Classification Report:
      precision    recall  f1-score   support

     0       0.78      0.73      0.75         211
     1       0.74      0.79      0.77         209

 accuracy                   0.76         420
 macro avg                   0.76      0.76      0.76         420
 weighted avg                 0.76      0.76      0.76         420

Cluster 0 - Confusion Matrix (normalized):
[[0.72985782 0.27014218]
 [0.20574163 0.79425837]]
Cluster 0 - RMSE: 0.4879500364742666

```

Рисунок 3.14 – Метрики для cluster 0

```

Cluster 1 - Classification Report:
      precision    recall  f1-score   support

     0       0.73      0.62      0.67          91
     1       0.71      0.80      0.75         106

 accuracy                   0.72         197
 macro avg                   0.72      0.71      0.71         197
 weighted avg                 0.72      0.72      0.71         197

Cluster 1 - Confusion Matrix (normalized):
[[0.61538462 0.38461538]
 [0.19811321 0.80188679]]
Cluster 1 - RMSE: 0.5331641017462287

```

Рисунок 3.15 – Метрики для cluster 1

```

Summary of results for all clusters
Cluster 0:
True positive rate = 0.73
False positive rate = 0.27
True negative rate = 0.794
False negative rate = 0.206
RMSE: 0.4879500364742666

Cluster 1:
True positive rate = 0.615
False positive rate = 0.385
True negative rate = 0.802
False negative rate = 0.198
RMSE: 0.5331641017462287

```

Рисунок 3.16 – Метрики для мережі довгої короткочасної пам'яті

Виходячи з отриманих результатів, можна зробити висновок, що модель Cluster 0 демонструє хороші результати, 73% при передбаченні позитивних результатів, та 79% при передбаченні негативних. RMSE знаходиться також у межах (0; 0.5) що свідчить про невисокий рівень помилок.

### 3.6 Реалізація алгоритму Random Forest

Для реалізації випадкового лісу вирішено встановити значення `n_estimators` на значення 200, тобто алгоритм буде 200 дерев рішень, що дозволяє отримати стабільніші результати. Для генератора випадкових чисел встановлено фіксоване початкове значення 64, щоб результати були відтворюваними при кожному запуску.

Розглянемо отримані результати на рисунках 3.17, 3.18, 3.19, 3.20, 3.21.

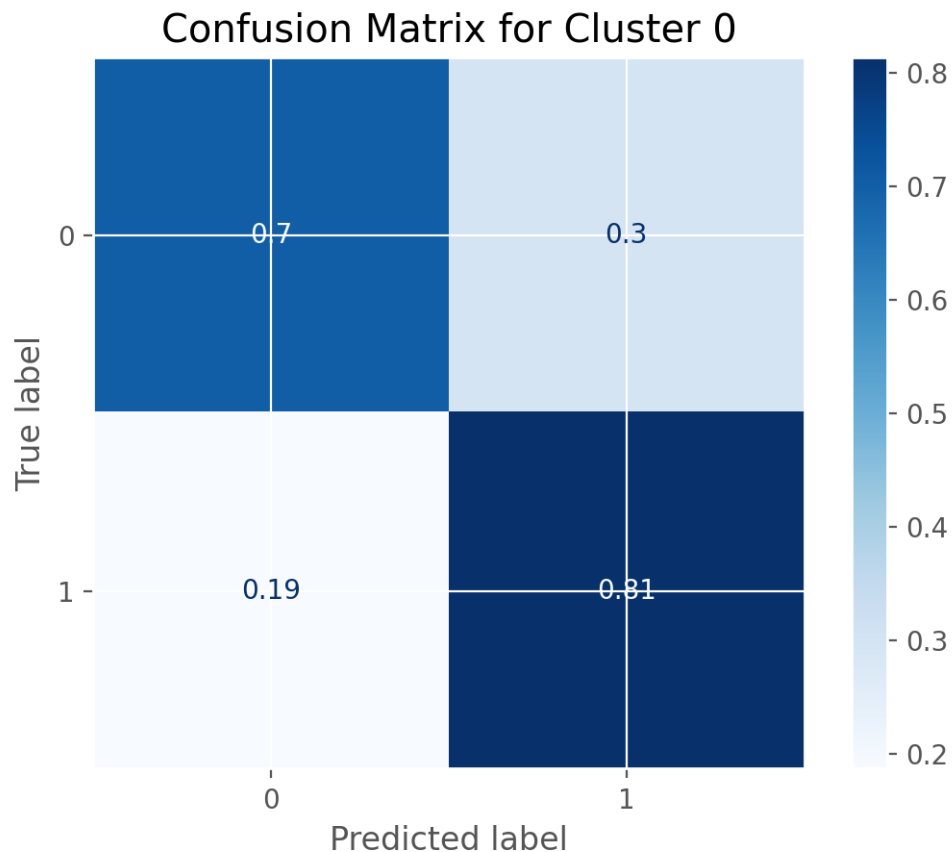


Рисунок 3.17 – Результат для кластеру 0

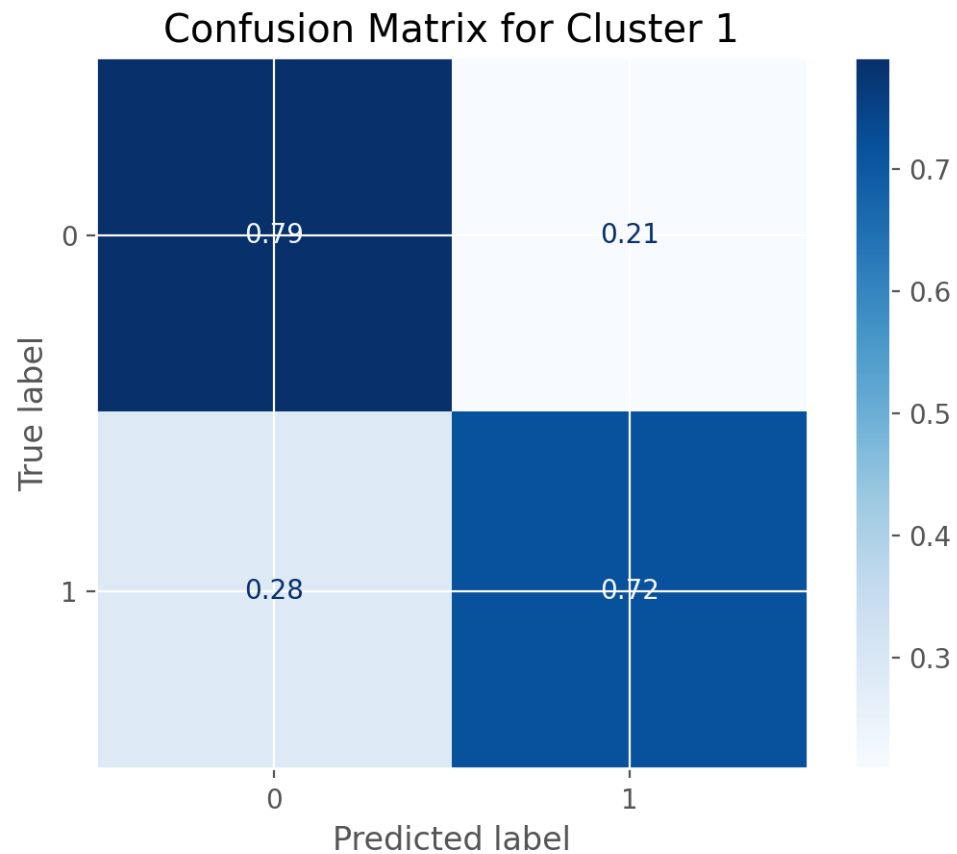


Рисунок 3.18 – Результат для кластеру 1

```

Training Random Forest model for cluster 0
Cluster 0 - Classification Report:
              precision    recall  f1-score   support

     0         0.79         0.70         0.74         211
     1         0.73         0.81         0.77         209

 accuracy         0.76         0.76         0.76         420
 macro avg         0.76         0.76         0.76         420
 weighted avg         0.76         0.76         0.76         420

Cluster 0 - Confusion Matrix (normalized):
[[0.7014218  0.2985782 ]
 [0.18660287 0.81339713]]
Cluster 0 - RMSE: 0.49280538030458115

```

Рисунок 3.19 – Метрики для cluster 0

```

Training Random Forest model for cluster 1
Cluster 1 - Classification Report:
      precision    recall  f1-score   support

     0       0.71      0.79      0.75         91
     1       0.80      0.72      0.76        106

 accuracy          0.75         197
 macro avg          0.75         197
 weighted avg       0.76         197

Cluster 1 - Confusion Matrix (normalized):
[[0.79120879 0.20879121]
 [0.28301887 0.71698113]]
Cluster 1 - RMSE: 0.4987293499153675

```

Рисунок 3.20 – Метрики для cluster 1

```

Summary of results for all clusters:
Cluster 0:
True positive rate = 0.701
False positive rate = 0.299
True negative rate = 0.813
False negative rate = 0.187
RMSE: 0.49280538030458115

Cluster 1:
True positive rate = 0.791
False positive rate = 0.209
True negative rate = 0.717
False negative rate = 0.283
RMSE: 0.4987293499153675

```

Рисунок 3.21 – Метрики для алгоритму Random Forest

Згідно отриманих результатів RMSE для обох кластерів менше 0.5. Для обох кластерів модель показує збалансовані результати, з хорошими значеннями recall та precision.

### 3.7 Реалізація рандомного торгового механізму

Для визначення чи результати методів та метрик мають якийсь сенс, порівняємо їх із результатами рандомного торгового механізму. Цей механізм

генерує випадкові рішення при прогнозуванні. Результати можна побачити на рисунках 3.22, 3.23, 3.24.

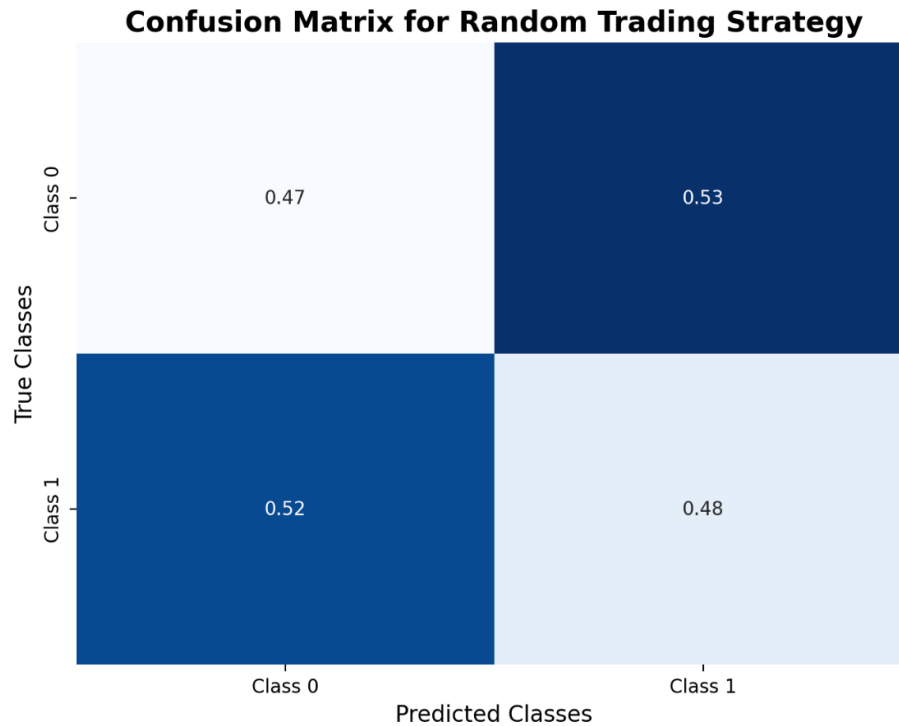


Рисунок 3.22 – Результат для рандомного торгового механізму

```

Random Trading Strategy - Classification Report:
              precision    recall  f1-score   support

     0         0.45      0.47      0.46         285
     1         0.50      0.48      0.49         315

 accuracy          0.47          0.47          0.47         600
 macro avg         0.47          0.47          0.47         600
 weighted avg     0.47          0.47          0.47         600

```

```

Random Trading Strategy - Confusion Matrix (normalized):
[[0.47017544 0.52982456]
 [0.52380952 0.47619048]]
Random Trading Strategy - RMSE: 0.725718035235908

```

Рисунок 3.23 – Метрики для рандомного торгового механізму

```

Summary of results for Random Trading Strategy:
True positive rate = 0.47
False positive rate = 0.53
True negative rate = 0.476
False negative rate = 0.524
RMSE: 0.725718035235908

```

Рисунок 3.24 – Загальний результат для рандомного торгового механізму

### 3.8 Висновки до розділу 3

Сьогодні фінансові ринки стають все доступнішими для звичайних користувачів, тому виникає необхідність знайти більш ефективний інструмент аналізу та прогнозування для подальшого інвестування.

У цій роботі здійснено порівняльний аналіз методів та моделей прогнозування на фондових ринках при різній волативності даних, та вивчено їх результати.

Під час аналізу були оцінені результати трьох методів моделювання: багатошарової нейронної мережі (DNN), мережі довгої короткочасної пам'яті (LSTM) та алгоритму Random Forest. Дані були поділені на два кластери в залежності від волативності акцій, що дало змогу оцінити ефективність кожного методу в різних умовах. Також для порівняння вирішено використати рандомний торговий механізм. Таблиця з значеннями основних метрик наведена у таблиці 3.1.

Таблиця 3.1 – Таблиця метрик розглянутих методів

Метод	Номер кластера	Точність (Accuracy)	Precision (середнє)	Recall (середнє)	F1-score	RMSE
DNN	0	0,75	0,76	0,75	0,75	0,4976
	1	0,77	0,78	0,77	0,77	0,4832
LSTM	0	0,76	0,76	0,76	0,76	0,4879
	1	0,72	0,72	0,71	0,71	0,5332
Random Forest	0	0,76	0,76	0,76	0,76	0,4928
	1	0,75	0,75	0,75	0,75	0,4987
Рандомний торговий показник		0,47	0,47	0,47	0,47	0,7257

Порівнюючи результати моделей та методів із рандомним торговим механізмом отримуємо підтвердження, що отримані результати мають сенс.



Рандомний торговий механізм програє у метриках усім трьом методам показуючи високий показник RMSE та низькі показники для таких метрик, як precision та recall.

Згідно отриманих результатів DNN показує найкращі результати в кластері 1, де дані характеризуються нижчою волативністю. Цей метод забезпечує високий рівень точності, що відображається у найвищих показниках серед розглянутих метрик, а також має найменшу середньоквадратичну похибку (RMSE). Це робить DNN найбільш ефективним методом для прогнозування цін закриття в стабільних ринкових умовах.

У свою чергу LSTM, навпаки, є найкращим вибором для кластеру 0, що характеризується високою волативністю. Ця модель показує найкращі результати за всіма метриками у кластері 0, а також найменшу похибку RMSE. Однак у кластері 1 її ефективність знижується через вищий рівень хибно позитивних передбачень, що обмежує її універсальність.

Random Forest показує стабільні, але середні результати в обох кластерах. Цей метод є універсальним, однак він поступається нейронним мережам у точності розпізнавання складних закономірностей. Він демонструє хороші показники у кластері 0, які можна порівняти з результатами мережі LSTM, проте значення RMSE вищі. У кластері 1 даний метод програє DNN за точністю та узгодженістю передбачень.

Таким чином, багатошарова нейронна мережа показує найкращі результати при стабільних умовах, тоді як для більш волативних ринків оптимальним є підхід із використанням LSTM. Random Forest можна застосовувати як базову модель, однак для досягнення найкращих результатів доцільно віддавати перевагу нейронним мережам [33].

## РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ

Сучасний фінансовий ринок характеризується високою динамічністю, що ускладнює прогнозування цін на біржі. Ефективне управління інвестиційними портфелями, прийняття рішень щодо купівлі або продажу активів, а також розробка стратегій управління ризиками повинні базуватись на точних прогнозах поведінки ринкових показників, таких як ціни акцій. Традиційні методи аналізу та прогнозування рідко забезпечують високу точність через нелінійність, непередбачуваність і велику кількість факторів, які здатні на це вплинути.

У зв'язку з цим все більше уваги приділяється використанню методів штучного інтелекту, зокрема нейронних мереж, здатних моделювати складні взаємозв'язки у фінансових даних. Нейронні мережі мають здатність навчатися на великих обсягах даних і можуть автоматично виявляти приховані закономірності в часових рядах, що робить їх потужним інструментом для прогнозування цін акцій.

Виходячи з вище сказаного, даний стартап можна вважати актуальним та таким, який має шанс та здатний зайняти свою нішу на ринку.

### 4.1 План розробки стартапу та масштабування його на ринок

Наведемо план розробки стартапу, спрямованого на розробку підходу щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж, а також заходи щодо його масштабування на ринку.

На початковому етапі необхідно провести маркетинговий аналіз, який включає:

- 1) конкурентний аналіз переваг та недоліків існуючих моделей і систем на ринку, зокрема традиційних економетричних моделей та сучасних алгоритмів машинного навчання;
- 2) формування ідеї стартапу, визначення основних категорій користувачів та розуміння їх потреб;
- 3) розробка стратегії виходу на ринок на основі аналізу ринкової ситуації.

Наступний етап передбачає створення структури та планування всіх етапів запуску, який включає:

- 1) складання детального плану розвитку та побудови таймлайну, який включає основні етапи розробки продукту – від підготовки даних та розробки моделі нейронної мережі до її тестування та впровадження на ринку;
- 2) оцінку ресурсів і обсягів виробництва необхідних для створення та підтримки системи;
- 3) розрахунок необхідних витрат на розробку та запуск стартапу.

Далі необхідно провести детальний аналіз фінансових показників та ризиків. На даному етапі потрібно провести.

1. Оцінку стартових витрат на розробку, маркетинг та операційну діяльність.
2. Розрахунок ключових фінансових показників (собівартість продукту, встановлення ціни на послуги, розрахунок податкових зобов'язань та чистого прибутку) та оцінка рентабельності продажів і визначення періоду окупності.
3. Оцінку та розробку стратегії мінімізації ризиків проекту, таких як зміни в економічному середовищі, непередбачуваність ринків акцій або конкуренція з боку великих гравців. Завершальним етапом буде комерціалізація та масштабування продукту.

Для успішного виведення продукту на ринок та його подальшого масштабування потрібно:

- 1) вивчити фінансовий ринок з метою залучення інвесторів, які зацікавлені в інноваційних рішеннях для фінансового прогнозування;
- 2) створити детальний план для інвесторів, що включає опис поточних можливостей продукту, оцінку ринкового потенціалу та шляхи подальшого розвитку;
- 3) вибрати канали для презентації стартапу та комунікації з потенційними інвесторами.

Далі наведемо результати виконання кожного з цих етапів.

#### 4.2 Опис ідеї стартап-проекту

Стартап-проект спрямований на розробку підходу щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж. Метою проекту є створення автоматизованої моделі, здатної з високою точністю передбачати майбутні зміни цін акцій на основі історичних даних.

У таблиці 4.1 наведена інформаційна карта стартапу.

Таблиця 4.1 – Інформаційна карта стартап-проекту

Назва проекту	PredictStock
Автори проекту	Гирила Степан Ігорович
Коротка анотація	Система використовує нейронні мережі для прогнозування вартості акцій на основі історичних даних по закриттю біржового ринку
Термін реалізації проекту	12 місяців

Продовження таблиці 4.1

Необхідні ресурси	<p>Офісне приміщення з доступом до Інтернету</p> <p>Сервери та спеціалізоване програмне забезпечення для обробки даних</p> <p>Фінансування для виплат зарплат, оренди приміщення, витрат на обчислювальні потужності, хмарні сервіси</p>
Опис проблеми, яку вирішує проект	<p>Проект дозволяє значно покращити точність прогнозування вартості акцій, що допоможе фінансовим аналітикам приймати зважені рішення</p>
Головні цілі та завдання проекту	<p>Метою є створення системи для прогнозування цін акцій з використанням машинного навчання та нейронних мереж</p>
Очікувані результати	<p>Розробка системи, що залучить фінансові компанії та аналітиків, створення масштабованої платформи для прогнозування акцій</p>

### 4.3 Технологічний аудит ідеї проекту

Наступним кроком є аналіз ідеї стартапу та проведення конкурентного аналізу. У таблиці 4.2 наведено опис ідеї стартапу.

Таблиця 4.2 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Основна ідея полягає у створенні системи, яка на основі історичних даних по закриттю біржового ринку прогнозує вартість акцій	Прогнозування вартості акцій	Користувач отримує прогноз вартості акцій для ухвалення інвестиційних рішень
	Постійне вдосконалення моделі	Система буде оновлюватись і точність прогнозу буде покращуватись

Далі проведемо порівняльний аналіз конкурентів проекту та наведемо результати у таблиці 4.3.

Таблиця 4.3 – Порівняльний аналіз конкурентів проекту

№ п/п	Техніко-економічні характеристики	Прогнозування на основі методу Random Forest	Прогнозування на основі мережі довгої короткочасної пам'яті (LSTM)	Прогнозування на основі багатопшарової нейронної мережі (DNN)	Власний проект
1	Точність прогнозу	Висока	Дуже висока	Середня	Залежить від вибору користувача
2	Швидкість обчислень	Дуже висока	Середня	Висока	Висока

Продовження таблиці 4.3

3	Складність використання	Середня	Висока	Низька	Низька
4	Вартість	Середня	Висока	Низька	Залежить від обраних технологій
5	Можливість персоналізації прогнозу	Середня	Висока	Обмежена	Висока
6	Наявність додаткових функцій	Обмежена	Широка	Мінімальна	Залежить від реалізації

Далі аналізуємо реальність технічно здійснити ідею проекту (таблиця 4.4).

Таблиця 4.4 – Технологічна здійсненність продукту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Створення підходу щодо прогнозування цін акцій за допомогою нейронних мереж	Python, TensorFlow, Keras	Наявні	Доступні
2	Побудова веб-інтерфейсу для взаємодії з користувачами	Streamlit, Plotly, HTML, CSS, JavaScript	Наявні	Доступні
3	Аналіз великих обсягів фінансових даних	Pandas, NumPy	Наявні	Доступні
Обрана технологія реалізації ідеї проекту: Python. Фреймворк: Streamlit.				

#### 4.4 Аналіз ринкових можливостей запуску стартап-проекту

Далі проведемо попередній аналіз ринку для запуску стартап-проекту (таблиця 4.5).

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн/ум.од	7000
3	Динаміка ринку (якісна оцінка)	Позитивна, ринок швидко зростає завдяки інтересу до автоматизованого прогнозування фінансових ринків
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Необхідна відповідність міжнародним стандартам безпеки даних та конфіденційності
6	Середня норма рентабельності в галузі (або по ринку), %	В залежності від типу і галузі акцій може варіюватись від 6% до 15%



Тепер проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені в проекті (таблиця 4.6).

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреби, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Прогнозування цін акцій для ухвалення інвестиційних рішень	Приватні інвестори	Зацікавленість у простоті використання та зрозумілості інтерфейсу	Точність прогнозів та швидкість обробки даних
2	Автоматизація фінансових аналітичних процесів	Фінансові компанії, банки, інвестиційні компанії	Орієнтовані на надійність та гнучкість моделей для довгострокового прогнозування	Інтеграція з існуючими фінансовими системами, персоналізація прогнозів
3	Підтримка рішень для торгових стратегій	Хедж-фонди, трейдери	Потреба у швидких прогнозах для короткострокових рішень	Висока швидкість обчислень та реалістичність прогнозів для швидких операцій

Обрахуємо фактори загроз (таблиця 4.7) та можливостей (таблиця 4.8). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску продукту на ринок. Фактори можливостей же треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 4.7 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Присутність великих компаній, що вже мають своїх користувачів і репутацію	Створення унікальних моделей прогнозування, які відрізняються точністю та персоналізацією
2	Технологічна складність	Невизначеність в ефективності прогнозів у складних ринкових умовах	Впровадження різних моделей для коротко- і довгострокових прогнозів
3	Підтримка користувачів	Високі очікування щодо точності прогнозів і швидкості отримання результатів	Постійне вдосконалення моделей та надання оперативної технічної підтримки
4	Регуляторні обмеження	Можливі обмеження або зміни у фінансовому регулюванні для подібних продуктів	Проведення правового аналізу ринку та адаптація до вимог законодавства

Таблиця 4.8 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростаючий ринок	Інтерес до автоматизованого прогнозування зростає серед інвесторів та трейдерів	Швидкий вихід на ринок та активний маркетинг для захоплення частки ринку
2	Інтеграція з платформами	Можливість інтеграції продукту з іншими фінансовими платформами	Розробка API для легкої інтеграції з існуючими фінансовими системами
3	Адаптивність моделей	Можливість персоналізованого прогнозування для різних сегментів ринку	Налаштування моделей під специфічні потреби різних типів користувачів
4	Вартість запуску	Доступність технологій дозволяє створити конкурентоспроможний продукт за низькою вартістю	Вихід на ринок з конкурентними цінами або безкоштовним MVP для залучення нових користувачів

Далі розглянемо питання конкуренції, а саме визначимо її тип та рівень (таблиця 4.9).

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції: недосконала конкуренція	На ринку є обмежена кількість подібних продуктів, але з різним рівнем точності прогнозів	Активна реклама застосунку, розробка моделей прогнозування для залучення цільової аудиторії
2. За рівнем конкурентної боротьби: міжнародний	Є проекти з міжнародною аудиторією, доступні на глобальних платформах	Розширення ринку через створення багатомовного інтерфейсу і вихід на міжнародні платформи
3. За галузевою ознакою: внутрішньогалузева	Конкуренція в рамках ринку фінансового прогнозування	Забезпечення різних варіантів прогнозування для різних сегментів користувачів
4. Конкуренція за видами товарів: товарно-родова	Інші системи для фінансового аналізу пропонують подібний функціонал	Покращення якості прогнозування і розширення функцій для фінансових аналітиків і трейдерів
5. За характером конкурентних переваг: нецінова	Конкуренти зосереджуються на якості прогнозів, зручності використання	Розробка якісніших моделей та інтуїтивно зрозумілого інтерфейсу, що забезпечить конкурентну перевагу
6. Інтенсивність конкуренції: марочна	Є сильні бренди, які вже мають свою клієнтську базу	Вибудова впізнаваного бренду через ефективну комунікаційну та маркетингову стратегію

Далі необхідно виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 4.10).

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамітники
	Інші системи прогнозування фінансових даних	Нові стартапи в сфері прогнозування акцій	Використання готових моделей та даних	Користувачі системи прогнозування акцій	Інші системи, що прогнозують ринки акцій
Висновки	Невисока інтенсивність конкуренції, є потенціал зростання	Ризик входження нових стартапів на ринок	Незалежність від постачальників	Клієнти не мають значного впливу на ціноутворення	Товарозамітники відсутні

Маючи результати аналізу конкуренції (таблиця 4.10), характеристики ідеї стартап-проекту (таблиця 4.5), характеристики потенційних клієнтів і їх вимоги до продукту (таблиця 4.6) та фактори ринкового середовища (таблиці 4.7 і 4.8) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (таблиця 4.11).

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Точність прогнозування	Забезпечення точних прогнозів за допомогою комплексних моделей
2	Швидкість обчислень	Використання сучасних технологій для пришвидшення процесів обчислення і видачі прогнозів
3	Простота у використанні	Інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко завантажувати дані та отримувати результати
4	Персоналізація	Можливість налаштування прогнозів під конкретні потреби клієнтів, включаючи вибір моделей і параметрів
5	Безкоштовний доступ до MVP	Забезпечення безкоштовного доступу на початковому етапі для швидкого залучення клієнтів та тестування продукту

Тепер можна провести аналіз сильних та слабких сторін продукту (таблиця 4.12).

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін підходу

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	1	2	3
1	Точність прогнозування	15			+				
2	Швидкість обчислень	16					+		
3	Простота у використанні	18				+			
4	Персоналізація	17					+		
5	Безкоштовний доступ до MVP	20							+

Далі проведемо SWOT-аналіз продукту (таблиця 4.13).

Таблиця 4.13 – SWOT-аналіз стартап-проекту

Сильні сторони	Слабкі сторони	Можливості	Загрози
Точність прогнозування	Відсутність сильного бренду	Покращення точності прогнозування	Поява нових конкурентних продуктів
Швидкість обчислень	Не сформована клієнтська база	Інтеграція з торговими платформами	Збільшення конкуренції на міжнародному рівні
Простота у використанні	Низька впізнаваність на ринку	Розширення функціональності продукту	Технологічні бар'єри, які можуть вплинути на швидкість впровадження

Продовження таблиці 4.13

Персоналізація	Відсутність додаткових каналів маркетингу	Впровадження персоналізованих рішень	Підвищення очікувань користувачів від систем прогнозування
Безкоштовний доступ до MVP	Недостатня кількість інтеграцій з іншими системами		

Завдяки проведенню SWOT-аналізу, ми змогли визначити сильні та слабкі сторони, можливості та загрози, пов'язані з конкуренцією та плануванням стартап-проекту. Далі спроектуємо альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок та наведемо результати у таблиці 4.14.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Вихід на ринок з базовими функціями прогнозування	80%	5 місяців
2	Пропонування користувачам обмеженого безкоштовного доступу	75%	4 місяці
3	Представлення продукту з інтеграцією в торгові платформи	55%	7 місяці



У даному пункті був проведений детальний аналіз ринку та продукту. Також відповідно до результатів проведеного конкурентного аналізу, визначених факторів ринку та його сприятливість, описання ідеї та характеристик стартап-проекту, робимо висновок , що існують дуже сприятливі умови для виходу продукту на ринок.

#### 4.5 Розроблення ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (таблиця 4.15).

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит у межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Персональні користувачі	Висока	30%	Середня	Висока
2	Малі компанії, що займаються аналізом ринкових даних	Середня	25%	Середня	Середня

Продовження таблиці 4.15

3	Державні установи, що займаються аналізом економічних показників	Низька	15%	Низька	Низька
4	Фінансові установи	Середня	20%	Висока	Середня
Які цільові групи обрано: 1, 2					

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Персональні користувачі та малі компанії	Диференційованого маркетингу	Універсальність продукту та висока якість аналізу даних	Масштабування і оптимальні витрати

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.17, 4.18).

Таблиця 4.17 – 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*
Ні	Так	Ні	Виклику лідера

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Універсальність Простота у використанні Якість аналізу ринкових даних та прогнозування	Оптимальних витрат	Універсальність Простота у використанні Висока якість результатів Безкоштовне використання при MVP	Продукт, що легко інтегрується з іншими системами Зручний для користувачів Забезпечує високоточний аналіз даних

#### 4.6 Розроблення маркетингової програми стартап-проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 4.19) та побудувати концепцію маркетингових комунікацій (таблиця 4.20).

Таблиця 4.19 – Ключові переваги концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Прогнозування цін акцій	Висока точність прогнозування на основі історичних даних	Використання сучасних нейронних мереж для підвищення точності
2	Адаптивність	Можливість адаптації моделі до змін у ринкових умовах	Гнучкість алгоритмів, що дозволяє швидко реагувати на новини
3	Зручність використання	Легкий у використанні інтерфейс для кінцевих користувачів	Інтуїтивно зрозумілий інтерфейс, що не вимагає спеціальних знань

Таблиця 4.20 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламно о повідомле ння	Концепц ія рекламн ого зверненн я
1	Пошук надійних інструменті в для інвестицій	B2B-продажі, фінансові платформи, спеціалізовані конференції	Надійність, точність, адаптивність	Інформува ти про точність прогнозува ння та адаптивніс ть продукту	Підкресл ити унікальн ість моделі прогнозу вання, що забезпеч ує переваги
2	Пошук ефективних рішень для трейдингу	Соціальні мережі, фінансові блоги, таргетована реклама	Доступність, простота використання, безкоштовне MVP	Сформував и уявлення про легкість використа ння системи	Реклама в лідерів думок, поясню ючи простоту інтеграці ї в робочий процес

#### 4.7 Висновки до розділу 4

Цей розділ був присвячений детальному вивченню стартап-проекту, спрямованого на розробку підходу щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж. У розділі проаналізовано ключові аспекти розробки стратегії виходу на ринок і маркетингової стратегії даного продукту.

В результаті проведеної роботи було виявлено, що ринок для нашого проекту є досить перспективним, оскільки є лише декілька конкурентів, які не покривають усіх можливостей та функціональних вимог сучасних інвесторів. Наш підхід забезпечує високу точність прогнозування курсів акцій, адаптивність до змін ринкової кон'юнктури та зручний інтерфейс, що відкриває можливість стати лідером у цій ніші.

Було опрацьовано сильні та слабкі сторони проекту, проведено SWOT-аналіз, а також аналіз конкурентного середовища та цільової аудиторії. На основі всіх отриманих даних була сформульована концепція маркетингової стратегії, яка приділяє увагу ключовим групам потенційних споживачів, які можуть отримати найбільшу користь від використання нашої системи.

Таким чином, стартап-проект має всі шанси успішно вийти на ринок, оскільки пропонує інноваційний та універсальний підхід до прогнозування цін на акції, який здатний задовольнити потреби сучасних інвесторів.

## ВИСНОВКИ

У дисертації були проведені дослідження щодо залежності якості прогнозування вартості акцій із використанням нейронних мереж від волативності даних про ціни акцій.

В рамках роботи було оглянуто такі популярні нейронні мережі як багатошарова нейронна мережа та мережа довгої короткочасної пам'яті, а також алгоритм Random Forest. Також проведено роботу з рандомним торговим показником. Після реалізації цих методів було проведено аналіз отриманих результатів з метою оцінки їхньої ефективності при прогнозуванні цін акцій в умовах різної волативності.

Зроблено висновок, що дані методи залежні від волативності вхідних даних та показують різний результат для різних кластерів, на які необхідно зважати при виборі конкретного інструменту для прогнозування цін, оскільки не існує універсального рішення для всіх можливих задач.

Було розроблено стратегію стартап-проекту та його комерційного використання, досліджено ринкові можливості запуску, визначено фактори загроз та можливостей, проведений аналіз конкурентів, описана ринкова стратегія, сформовані слабкі та сильні сторони та виготовлена концепція маркетингових комунікацій.

В цілому, дане дослідження підтверджує, що нейронні мережі та алгоритми, такі як Random Forest, здатні допомогти у роботі інвесторам та трейдерам, даючи можливість прогнозувати вартість акцій по закриттю біржового ринку.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Murphy J. J. Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York: Penguin, 1999. 150 p. URL: [https://www.google.com.ua/books/edition/Technical\\_Analysis\\_of\\_the\\_Financial\\_Mark/teitAAAAQBAJ](https://www.google.com.ua/books/edition/Technical_Analysis_of_the_Financial_Mark/teitAAAAQBAJ) (Last accessed: 10.09.2024).
2. Hayes A. Technical Analysis: What It Is and How to Use It in Investing. Investopedia. URL: <https://www.investopedia.com/terms/t/technicalanalysis.asp#citation-1> (Last accessed: 11.09.2024).
3. Baker H. K. Equity Markets, Valuation, and Analysis. Hoboken: Wiley, 2020. 119 p. URL: [https://www.google.com.ua/books/edition/Equity\\_Markets\\_Valuation\\_and\\_Analysis/lmnyDwAAQBAJ](https://www.google.com.ua/books/edition/Equity_Markets_Valuation_and_Analysis/lmnyDwAAQBAJ) (Last accessed: 11.09.2024).
4. Fundamental Analysis: Principles, Types, and How to Use It. Investopedia. URL: <https://www.investopedia.com/terms/f/fundamentalanalysis.asp#toc-understanding-fundamental-analysis> (Last accessed: 12.09.2024).
5. Box G. E. P., Jenkins G. M., Reinsel G. C., Time Series Analysis. Hoboken: John Wiley & Sons. 2008. 209 p. URL: <https://download.e-bookshelf.de/download/0003/8810/69/L-G-0003881069-0007953902.pdf> (Last accessed: 20.10.2024).
6. Hyndman R., Koehler A. B., Ord J. K., Snyder R. D., Forecasting with Exponential Smoothing: The State Space Approach. New York: Springer Science & Business Media. 2008. URL: [https://www.google.com.ua/books/edition/Forecasting\\_with\\_Exponential\\_Smoothing/GSyzoX8Lu9YC?hl=en&gbpv=0](https://www.google.com.ua/books/edition/Forecasting_with_Exponential_Smoothing/GSyzoX8Lu9YC?hl=en&gbpv=0) (Last accessed: 20.10.2024).



7. James G., Witten D., Hastie T., Tibshirani R., An Introduction to Statistical Learning. New York: Springer Science & Business Media. 2013. URL: [https://www.google.com.ua/books/edition/An\\_Introduction\\_to\\_Statistical\\_Learning/qcI\\_AAAAQBAJ](https://www.google.com.ua/books/edition/An_Introduction_to_Statistical_Learning/qcI_AAAAQBAJ) (Last accessed: 20.10.2024).
8. What are ARIMA models? IBM. URL: <https://www.ibm.com/topics/arima-model> (Last accessed: 12.09.2024).
9. A Brief History of the Neural Networks. KDnuggets. URL: <https://www.kdnuggets.com/a-brief-history-of-the-neural-networks> (Last accessed: 15.09.2024).
10. McCulloch W. S., Pitts W. A. Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. Chicago: The University of Chicago, 1943. 115 p. URL: <https://home.csulb.edu/~cwallis/382/readings/482/mcculloch.logical.calculus.ideas.1943.pdf> (Last accessed: 15.09.2024).
11. A Very Short History of Artificial Neural Networks. Medium. URL: <https://jimstone-68634.medium.com/a-very-short-history-of-artificial-neural-networks-9820dfd6d903> (Last accessed: 20.09.2024).
12. Understanding the McCulloch-Pitts (MP) neuron: A Fundamental Concept in Early Neural Network Development. Medium. URL: <https://medium.com/@marioarvind1790/understanding-the-mcculloch-pitts-mp-neuron-a-fundamental-concept-in-early-neural-network-c3a78c67f43e> (Last accessed: 20.09.2024).
13. Elbow Method for Optimal Cluster Number in K-Means. Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/> (Last accessed: 10.10.2024).
14. What is Silhouette Score? Medium. URL: <https://medium.com/@hazallgultekin/what-is-silhouette-score-f428fb39bf9a> (Last accessed: 10.10.2024).

15. An Introduction to the ReLU Activation Function. Built In. URL: <https://builtin.com/machine-learning/relu-activation-function> (Last accessed: 10.10.2024).
16. A Visual and Intuitive Guide to What Makes ReLU a Non-linear Activation Function. Daily Dose of Data Science. URL: <https://blog.dailydoseofds.com/p/a-visual-and-intuitive-guide-to-what> (Last accessed: 10.10.2024).
17. ELU Activation Function. Closeheat. URL: <https://closeheat.com/blog/elu-activation-function> (Last accessed: 10.10.2024).
18. Leaky ReLU Activation Function. Medium. URL: <https://medium.com/@akp83540/leaky-relu-activation-function-9fee1374f6bd> (Last accessed: 10.10.2024).
19. Randomized Leaky Rectified Linear Units. PaperWithCode. URL: <https://paperswithcode.com/method/rrelu> (Last accessed: 10.10.2024).
20. Understanding the Softmax Activation Function: A Comprehensive Guide. SingleStore. URL: <https://www.singlestore.com/blog/a-guide-to-softmax-activation-function/> (Last accessed: 12.10.2024).
21. Softmax Function: Advantages and Applications. Bot Penguin. URL: <https://botpenguin.com/glossary/softmax-function> (Last accessed: 12.10.2024).
22. Exploration 4.1: Multilayer Neural Networks. Classes. URL: <https://classes.engr.oregonstate.edu/eecs/fall2023/ai534-400/unit4/multilayer.html> (Last accessed: 12.10.2024).
23. Deep Learning 101: Lesson 9: Multi-layer Neural Network. Medium. URL: <https://muneebsa.medium.com/deep-learning-101-lesson-9-multi-layer-neural-network-7cd3a53066c8> (Last accessed: 12.10.2024).
24. Introduction to Long Short-Term Memory (LSTM). Simplilearn. URL: <https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/lstm> (Last accessed: 12.10.2024).

25. Long Short Term Memory Networks. Codecademy. URL: <https://www.codecademy.com/article/long-short-term-memory-networks> (Last accessed: 12.10.2024).
26. Random Forest Algorithm in Machine Learning. GeeksForGeeks. URL: <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/> (Last accessed: 12.10.2024).
27. Свічкові графіки. Керівництво для початківців. Binance Academy. URL: <https://academy.binance.com/uk/articles/a-beginners-guide-to-candlestick-charts> (Last accessed: 20.10.2024).
28. Осадчук П. Технічний аналіз криптовалют. Частина №1: індикаторний аналіз | Whitebase Trading №9. Друкарня. URL: <https://drukarnia.com.ua/articles/tekhnichnii-analiz-kriptovalyut-chastina-1-indikatornii-analiz-abo-whitebase-trading-9-ZfMdx> (Last accessed: 20.10.2024).
29. Python. URL: <https://www.python.org/downloads/> (Last accessed: 20.10.2024).
30. Jupyter Notebook. URL: <https://jupyter.org/> (Last accessed: 20.10.2024).
31. Yahoo Finance. Alphabet Inc. (GOOG). URL: <https://finance.yahoo.com/quote/GOOG/> (Last accessed: 20.10.2024).
32. What a Stock Split Is, Why Companies Do It, and How It Works, With an Example. Investopedia. URL: <https://www.investopedia.com/terms/s/stocksplit.asp> (Last accessed: 20.10.2024).
33. Гирила С.І., Савастьянов В.В. Підхід щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж. Збірник доповідей II Всеукраїнської науково-практичної конференції «Системні науки та інформатика», 25–29 листопада 2024 року, Київ. К., НН ІПСА КПІ ім. Ігоря Сікорського, 2024. С. 8. (дата звернення: 30.11.2024).

## ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import BatchNormalization
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, CSVLogger, EarlyStopping
from keras.optimizers import RMSprop, Adam, SGD, Nadam
from keras.layers import *
from keras.layers import Convolution1D, MaxPooling1D, Conv1D
from keras.layers import LSTM, GRU
from keras import regularizers
import seaborn as sns
# Data Imports
import yfinance as yf
import pandas as pd
import numpy as np
import random

from sklearn.preprocessing import MinMaxScaler
from pandas.tseries.offsets import BDay

# Visualization Imports
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
import scipy.stats as stats

# Neural Network Imports
import tensorflow as tf
from tensorflow.keras import models
from tensorflow.keras import regularizers
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Bidirectional
from tensorflow.keras.callbacks import ModelCheckpoint

SEED = 0
random.seed(SEED)
np.random.seed(SEED)
```

```

tf.random.set_seed(SEED)

pio.templates.default = "plotly_white"
%config InlineBackend.figure_format = 'retina'
sns.despine()
def highlight_half(data: pd.DataFrame, axis=1, precision=3):

    s = data.shape[1] if axis else data.shape[0]
    data_style = data.style.format(precision=precision)

    def apply_style(val):
        style1 = 'background-color: blue; color: white'
        style2 = 'background-color: red; color: white'
        return [style1 if x < s//2 else style2 for x in range(s)]

    display(data_style.apply(apply_style, axis=axis))
SYMBOL = "GOOG"
columns = ['open', 'high', 'low', 'close']

ticker = yf.Ticker(SYMBOL)

end_date = "2024-10-15"

df = ticker.history(start="2017-01-01", end=end_date)
df.columns = df.columns.str.lower()

show(df)
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

sse = []
sil_scores = []
cluster_range = range(2, 10)

for n_clusters in cluster_range:
    kmeans = KMeans(n_clusters=n_clusters, random_state=0)
    kmeans.fit(scaled_volatilities)

    sse.append(kmeans.inertia_)

    score = silhouette_score(scaled_volatilities, kmeans.labels_)
    sil_scores.append(score)

fig, ax1 = plt.subplots(figsize=(10, 6))

ax1.set_xlabel('Number of Clusters')

```

```
ax1.set_ylabel('Sum of Squared Distances (SSE)', color='tab:blue')
ax1.plot(cluster_range, sse, marker='o', linestyle='-', color='tab:blue', label='SSE')
ax1.tick_params(axis='y', labelcolor='tab:blue')
```

```
ax2 = ax1.twinx()
ax2.set_ylabel('Silhouette Score', color='tab:orange')
ax2.plot(cluster_range, sil_scores, marker='o', linestyle='-', color='tab:orange',
label='Silhouette Score')
ax2.tick_params(axis='y', labelcolor='tab:orange')
```

```
plt.title('Elbow Method and Silhouette Score for Optimal K')
fig.tight_layout()
plt.show()
```

Score

```
optimal_clusters = cluster_range[np.argmax(sil_scores)]
print(f"Optimal number of clusters (based on silhouette score): {optimal_clusters}")
```

```
kmeans = KMeans(n_clusters=optimal_clusters, random_state=0)
clusters = kmeans.fit_predict(scaled_volatilities)
```

```
X_clustered = {}
```

```
Y_clustered = {}
```

```
for i, cluster in enumerate(clusters):
```

```
    if cluster not in X_clustered:
```

```
        X_clustered[cluster] = []
```

```
        Y_clustered[cluster] = []
```

```
    X_clustered[cluster].append(X[i])
```

```
    Y_clustered[cluster].append(Y[i])
```

```
def shuffle_in_unison(a, b):
```

```
    assert len(a) == len(b)
```

```
    permutation = np.random.permutation(len(a))
```

```
    return a[permutation], b[permutation]
```

```
def create_Xt_Yt(X, Y, percentage=0.8):
```

```
    p = int(len(X) * percentage)
```

```
    X_train = X[:p]
```

```
    Y_train = Y[:p]
```

```
    X_test = X[p:]
```

```
    Y_test = Y[p:]
```

```
    return X_train, X_test, Y_train, Y_test
```

```
X_train_clustered, X_test_clustered, Y_train_clustered, Y_test_clustered = {}, {}, {}, {}
```

```
for cluster in X_clustered.keys():
```

```
    X_arr = np.array(X_clustered[cluster])
```

```

Y_arr = np.array(Y_clustered[cluster])

X_arr, Y_arr = shuffle_in_unison(X_arr, Y_arr)

X_train_clustered[cluster], X_test_clustered[cluster], Y_train_clustered[cluster],
Y_test_clustered[cluster] = create_Xt_Yt(X_arr, Y_arr)
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix, mean_squared_error,
ConfusionMatrixDisplay
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, BatchNormalization, ELU, Dropout, Activation
from tensorflow.keras.optimizers.legacy import Nadam
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint
from tensorflow.keras import regularizers

opt = Nadam(learning_rate=0.002)
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.9, patience=25,
min_lr=0.000001, verbose=1)

cluster_results = {}

for cluster in X_train_clustered.keys():
    print(f"Training model for cluster {cluster}")

    model = Sequential()
    model.add(Dense(64, input_dim=30, activity_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    model.add(ELU())
    model.add(Dropout(0.4))
    model.add(Dense(32, activity_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    model.add(ELU())
    model.add(Dense(2))
    model.add(Activation('softmax'))

    model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

    checkpoint_path = f"model_cluster_{cluster}.hdf5"
    checkpointer = ModelCheckpoint(filepath=checkpoint_path, verbose=1, save_best_only=True)

    history = model.fit(
        np.array(X_train_clustered[cluster]), np.array(Y_train_clustered[cluster]),
        epochs=150,
        batch_size=128,

```

```

        verbose=1,
        validation_data=(np.array(X_test_clustered[cluster]),
np.array(Y_test_clustered[cluster])),
        callbacks=[reduce_lr, checkpointer],
        shuffle=True
    )
    model.load_weights(checkpoint_path)

    pred = model.predict(np.array(X_test_clustered[cluster]))

    pred_classes = np.argmax(pred, axis=1)
    true_classes = np.argmax(Y_test_clustered[cluster], axis=1)

    cluster_conf_matrix = confusion_matrix(true_classes, pred_classes)
    cluster_report = classification_report(true_classes, pred_classes)
    rmse = np.sqrt(mean_squared_error(true_classes, pred_classes))

    cluster_conf_matrix = cluster_conf_matrix / cluster_conf_matrix.astype(float).sum(axis=1,
keepdims=True)

    cluster_results[cluster] = {
        "confusion_matrix": cluster_conf_matrix,
        "classification_report": cluster_report,
        "rmse": rmse,
        "pred_classes": pred_classes,
        "true_classes": true_classes,
    }

    print(f"Cluster {cluster} - Classification Report:\n{cluster_report}")
    print(f"Cluster {cluster} - Confusion Matrix (normalized):\n{cluster_conf_matrix}")
    print(f"Cluster {cluster} - RMSE: {rmse}\n")

    plt.figure(figsize=(8, 6))
    ConfusionMatrixDisplay(cluster_conf_matrix).plot(cmap='Blues')
    plt.title(f"Confusion Matrix for Cluster {cluster}")
    plt.show()

    residuals = true_classes - pred_classes
    plt.figure(figsize=(10, 6))
    plt.hist(residuals, bins=np.arange(-1.5, 1.5, 0.5), color='blue', edgecolor='black')
    plt.title(f"Residuals Distribution for Cluster {cluster}")
    plt.xlabel("Residuals (True - Predicted)", fontsize=17, fontweight='bold')
    plt.ylabel("Frequency", fontsize=17, fontweight='bold')
    plt.xticks([-1, 0, 1], labels=["False Negative", "Correct", "False Positive"])
    plt.show()

```



```

plt.figure(figsize=(10, 6))
sns.kdeplot(residuals, fill=True, color="blue")
plt.title(f"Residuals Density Plot for Cluster {cluster}")
plt.xlabel("Residuals (True - Predicted)", fontsize=17, fontweight='bold')
plt.show()

print("Summary of results for all clusters:")
for cluster, results in cluster_results.items():
    print(f"Cluster {cluster}:")
    print(f"True positive rate = {round(results['confusion_matrix'][0][0], 3)}")
    print(f"False positive rate = {round(results['confusion_matrix'][0][1], 3)}")
    print(f"True negative rate = {round(results['confusion_matrix'][1][1], 3)}")
    print(f"False negative rate = {round(results['confusion_matrix'][1][0], 3)}")
    print(f"RMSE: {results['rmse']}\n")

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix, mean_squared_error,
ConfusionMatrixDisplay
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, BatchNormalization, LSTM, Dropout, LeakyReLU,
Activation
from tensorflow.keras.optimizers.legacy import Nadam
from tensorflow.keras.callbacks import ReduceLRonPlateau, ModelCheckpoint
from tensorflow.keras import regularizers

opt = Nadam(learning_rate=0.002)
reduce_lr = ReduceLRonPlateau(monitor='val_accuracy', factor=0.9, patience=25,
min_lr=0.000001, verbose=1)

cluster_results = {}

for cluster in X_train_clustered.keys():
    print(f"Training model for cluster {cluster}")

    model = Sequential()
    model.add(LSTM(units=32, input_shape=(30, 1),
                  return_sequences=False,
                  recurrent_dropout=0.9,
                  kernel_regularizer=regularizers.l2(0.01),
                  activity_regularizer=regularizers.l2(0.01)))
    model.add(Dropout(0.7))
    model.add(Dense(64))
    model.add(BatchNormalization())
    model.add(LeakyReLU())
    model.add(Dense(2))

```

```

model.add(Activation('softmax'))

model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

checkpoint_path = f"lstm_model_cluster_{cluster}.hdf5"
checkpointer = ModelCheckpoint(filepath=checkpoint_path, verbose=1, save_best_only=True)

X_train_lstm = np.array(X_train_clustered[cluster]).reshape((-1, 30, 1))
X_test_lstm = np.array(X_test_clustered[cluster]).reshape((-1, 30, 1))

history = model.fit(
    X_train_lstm, np.array(Y_train_clustered[cluster]),
    epochs=150,
    batch_size=128,
    verbose=1,
    validation_data=(X_test_lstm, np.array(Y_test_clustered[cluster])),
    callbacks=[reduce_lr, checkpointer],
    shuffle=True
)

model.load_weights(checkpoint_path)

pred = model.predict(X_test_lstm)

pred_classes = np.argmax(pred, axis=1)
true_classes = np.argmax(Y_test_clustered[cluster], axis=1)

cluster_conf_matrix = confusion_matrix(true_classes, pred_classes)
cluster_report = classification_report(true_classes, pred_classes)
rmse = np.sqrt(mean_squared_error(true_classes, pred_classes))

cluster_conf_matrix = cluster_conf_matrix / cluster_conf_matrix.astype(float).sum(axis=1,
keepdims=True)

cluster_results[cluster] = {
    "confusion_matrix": cluster_conf_matrix,
    "classification_report": cluster_report,
    "rmse": rmse,
    "pred_classes": pred_classes,
    "true_classes": true_classes,
}

print(f"Cluster {cluster} - Classification Report:\n{cluster_report}")
print(f"Cluster {cluster} - Confusion Matrix (normalized):\n{cluster_conf_matrix}")
print(f"Cluster {cluster} - RMSE: {rmse}\n")

```

```

plt.figure(figsize=(8, 6))
ConfusionMatrixDisplay(cluster_conf_matrix).plot(cmap='Blues')
plt.title(f"Confusion Matrix for Cluster {cluster}")
plt.show()

residuals = true_classes - pred_classes
plt.figure(figsize=(10, 6))
plt.hist(residuals, bins=np.arange(-1.5, 1.5, 0.5), color='blue', edgecolor='black')
plt.title(f"Residuals Distribution for Cluster {cluster}")
plt.xlabel("Residuals (True - Predicted)", fontsize=17, fontweight='bold')
plt.ylabel("Frequency", fontsize=17, fontweight='bold')
plt.xticks([-1, 0, 1], labels=["False Negative", "Correct", "False Positive"])
plt.show()

plt.figure(figsize=(10, 6))
sns.kdeplot(residuals, fill=True, color="blue")
plt.title(f"Residuals Density Plot for Cluster {cluster}")
plt.xlabel("Residuals (True - Predicted)", fontsize=17, fontweight='bold')
plt.show()

print("Summary of results for all clusters:")
for cluster, results in cluster_results.items():
    print(f"Cluster {cluster}:")
    print(f"True positive rate = {round(results['confusion_matrix'][0][0], 3)}")
    print(f"False positive rate = {round(results['confusion_matrix'][0][1], 3)}")
    print(f"True negative rate = {round(results['confusion_matrix'][1][1], 3)}")
    print(f"False negative rate = {round(results['confusion_matrix'][1][0], 3)}")
    print(f"RMSE: {results['rmse']}\n")
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, mean_squared_error,
ConfusionMatrixDisplay
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

cluster_results = {}

for cluster in X_train_clustered.keys():
    print(f"Training Random Forest model for cluster {cluster}")

    X_train = np.array(X_train_clustered[cluster])
    Y_train = np.array(Y_train_clustered[cluster])

    if X_train.ndim > 2:

```

```

    X_train = X_train.reshape(X_train.shape[0], -1)
if Y_train.ndim > 2:
    Y_train = Y_train.reshape(Y_train.shape[0], -1)
Y_train_classes = np.argmax(Y_train, axis=1)

rf_model = RandomForestClassifier(n_estimators=200, random_state=64)

rf_model.fit(X_train, Y_train_classes)

X_test = np.array(X_test_clustered[cluster])
Y_test = np.array(Y_test_clustered[cluster])

if X_test.ndim > 2:
    X_test = X_test.reshape(X_test.shape[0], -1)

true_classes = np.argmax(Y_test, axis=1)
pred_classes = rf_model.predict(X_test)

cluster_conf_matrix = confusion_matrix(true_classes, pred_classes)
cluster_report = classification_report(true_classes, pred_classes)
rmse = np.sqrt(mean_squared_error(true_classes, pred_classes))

cluster_conf_matrix = cluster_conf_matrix / cluster_conf_matrix.astype(float).sum(axis=1,
keepdims=True)

cluster_results[cluster] = {
    "confusion_matrix": cluster_conf_matrix,
    "classification_report": cluster_report,
    "rmse": rmse,
    "pred_classes": pred_classes,
    "true_classes": true_classes,
}

print(f"Cluster {cluster} - Classification Report:\n{cluster_report}")
print(f"Cluster {cluster} - Confusion Matrix (normalized):\n{cluster_conf_matrix}")
print(f"Cluster {cluster} - RMSE: {rmse}\n")

plt.figure(figsize=(8, 6))
ConfusionMatrixDisplay(cluster_conf_matrix).plot(cmap='Blues')
plt.title(f"Confusion Matrix for Cluster {cluster}")
plt.show()

residuals = true_classes - pred_classes
plt.figure(figsize=(10, 6))
plt.hist(residuals, bins=np.arange(-1.5, 1.5, 0.5), color='blue', edgecolor='black')
plt.title(f"Residuals Distribution for Cluster {cluster}")

```

```
plt.xlabel("Residuals (True - Predicted)", fontsize=17, fontweight='bold')
plt.ylabel("Frequency", fontsize=17, fontweight='bold')
plt.xticks([-1, 0, 1], labels=["False Negative", "Correct", "False Positive"])
plt.show()

plt.figure(figsize=(10, 6))
sns.kdeplot(residuals, fill=True, color="blue")
plt.title(f"Residuals Density Plot for Cluster {cluster}")
plt.xlabel("Residuals (True - Predicted)", fontsize=17, fontweight='bold')
plt.show()

print("Summary of results for all clusters:")
for cluster, results in cluster_results.items():
    print(f"Cluster {cluster}:")
    print(f"True positive rate = {round(results['confusion_matrix'][0][0], 3)}")
    print(f"False positive rate = {round(results['confusion_matrix'][0][1], 3)}")
    print(f"True negative rate = {round(results['confusion_matrix'][1][1], 3)}")
    print(f"False negative rate = {round(results['confusion_matrix'][1][0], 3)}")
    print(f"RMSE: {results['rmse']}\n")
```

## ДОДАТОК Б ДОПОВІДЬ У ВИГЛЯДІ ПРЕЗЕНТАЦІЇ

### Підхід щодо моделювання вартості акцій по закриттю біржового ринку із використанням нейронних мереж

Виконав:  
Студент групи КА-32мп  
Гирила С.І.  
Керівник:  
Савастьянов В.В.

**Мета роботи** — розробка підходу до прогнозування вартості акцій на закритті біржового ринку за допомогою нейронних мереж. Дослідження спрямоване на оцінку ефективності цих методів у різних умовах волатильності ринку.

**Об'єкт дослідження** — відкриті дані про торги акціями на фондовому ринку.

**Предмет досліджень** — методи кластеризації та прогнозування фінансових даних з використанням нейронних мереж і алгоритмів машинного навчання, а також оцінка їхньої точності за умов різної ринкової волатильності

## Актуальність роботи

Актуальність теми ґрунтується на низці ключових аспектів, які визначають необхідність такого дослідження.

По-перше, зростаюча волатильність фінансових ринків та обсяги даних вимагають нових інструментів для аналізу. Традиційні статистичні методи часто не враховують складні нелінійні залежності та великі обсяги історичних даних, що робить їх недостатньо ефективними у прогнозуванні вартості акцій. Нейронні мережі, зокрема рекурентні та глибокі, здатні ефективно моделювати складні взаємозв'язки між змінними, враховуючи як числові, так і категоріальні дані. Це відкриває можливості не лише для створення більш точних прогностичних моделей, але й для виявлення аномалій, розкриття прихованих закономірностей та адаптації моделей до різних ринкових умов.

По-друге, глобалізація фінансових ринків та їх взаємозалежність підвищують вимоги до якості прогнозування. Відсутність універсальних моделей для прогнозування фінансових показників вимагає розробки специфічних підходів, які враховують особливості конкретного ринку або фінансового інструменту. Неправильний вибір моделі або її недостатня адаптація до ринкових умов можуть призвести до значних втрат для інвесторів, компаній та інших учасників ринку.

По-третє, розробка методів прогнозування на основі нейронних мереж є важливим завданням як з наукової, так і з практичної точки зору. З наукової перспективи це сприяє розвитку методів інтелектуального аналізу даних та глибокого навчання. З практичного боку, це відкриває можливості для прийняття ефективних управлінських рішень, зменшення ризиків та підвищення конкурентоспроможності фінансових ринків.

3

## Технології, які були використані



4

## Вхідні дані

Для виконання нашої задачі було взято датасет з даними про акції компанії Alphabet з часу її виходу на IPO (першої емісії акцій) до сьогоднішнього дня. Тобто у період з 19 серпня 2004 року до 25 вересня 2024 року. Дані були взяті з ресурсу “Yahoo finance”. У цьому наборі даних присутні відомості про ціну акції на початку календарного дня, на кінець календарного дня, найвища ціна впродовж дня, найнижча ціна впродовж дня та кількість акцій, які були продані у цей день. Для своєї роботи я використовував дані про ціну акцій наприкінці дня. Ціна акцій у датасеті наведена у доларах США.

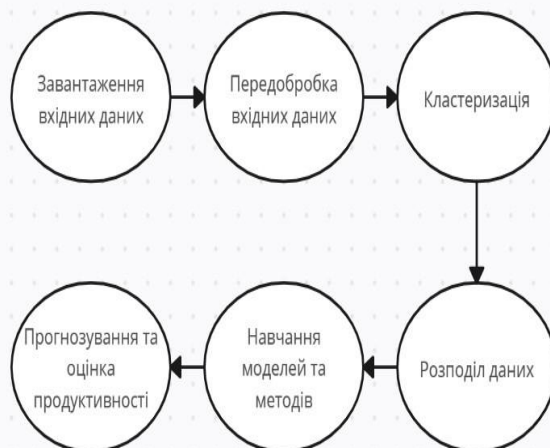
2024-09-13 00:00:00-04:00	156.362	159.275	156.110	158.370
2024-09-16 00:00:00-04:00	158.330	159.240	157.610	158.990
2024-09-17 00:00:00-04:00	160.090	161.580	159.410	160.280
2024-09-18 00:00:00-04:00	160.850	161.630	159.660	160.610
2024-09-19 00:00:00-04:00	164.820	164.980	162.522	163.240
2024-09-20 00:00:00-04:00	164.520	164.750	163.180	164.640
2024-09-23 00:00:00-04:00	165.340	166.610	162.950	163.070
2024-09-24 00:00:00-04:00	164.250	164.550	162.030	163.640
2024-09-25 00:00:00-04:00	162.970	164.217	162.775	162.990
2024-09-26 00:00:00-04:00	165.030	165.500	163.500	163.830
2024-09-27 00:00:00-04:00	163.910	166.970	163.830	165.290
2024-09-30 00:00:00-04:00	164.780	167.380	164.640	167.190
2024-10-01 00:00:00-04:00	168.860	170.440	165.900	168.420
2024-10-02 00:00:00-04:00	167.780	168.800	166.250	167.310
2024-10-03 00:00:00-04:00	165.820	167.910	165.370	167.210
2024-10-04 00:00:00-04:00	169.340	169.550	166.980	168.580
2024-10-07 00:00:00-04:00	169.140	169.900	164.130	164.390
2024-10-08 00:00:00-04:00	165.430	166.100	164.310	165.700
2024-10-09 00:00:00-04:00	164.855	166.280	161.120	163.060
2024-10-10 00:00:00-04:00	162.110	164.311	161.640	163.180
2024-10-11 00:00:00-04:00	163.330	165.270	162.500	164.520
2024-10-14 00:00:00-04:00	164.910	167.620	164.780	166.350

5

## Підготовка вхідних даних:

Відбувається кластеризація для сегментації даних на групи з подібними характеристиками волативності. Алгоритм K-Means групує всі точки даних у відповідні кластери, що дозволяє сегментувати фінансові дані за характеристиками їхньої волатильності.

Після кластеризації дані кожного кластера розділяються на навчальні та тестові вибірки у співвідношенні 80% для навчання і 20% для тестування. Щоб забезпечити випадковість, перед розділенням дані в межах кожного кластера перемішуються для запобігання зміщення моделі, викликаного нерівномірним розподілом даних.



6

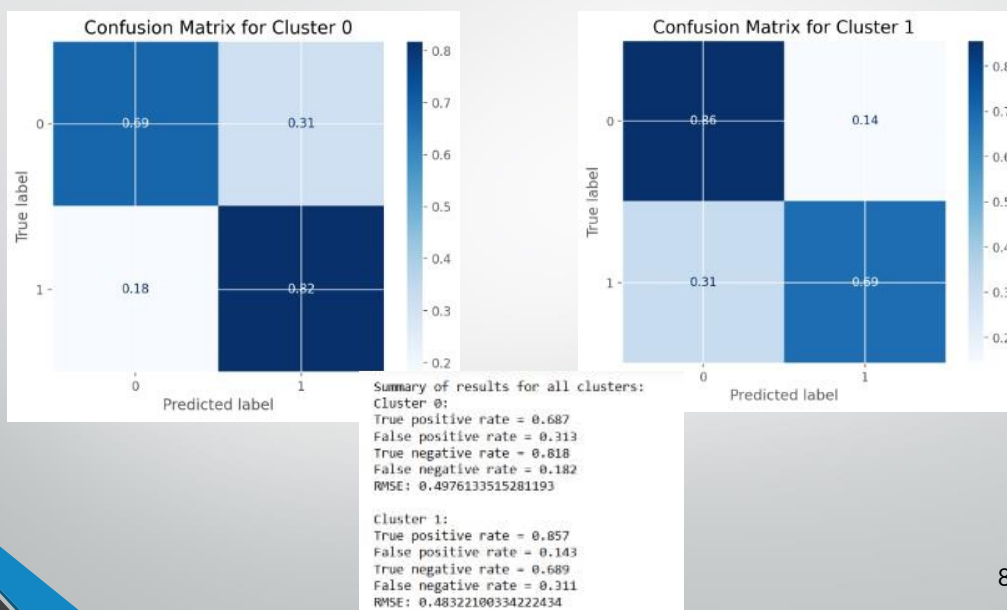


## Застосовані метрики

- True Positive Rate (TPR):
- $TPR = \frac{TP + FN}{TP}$
- False Positive Rate (FPR):
- $FPR = \frac{FP + TN}{FP}$
- True Negative Rate (TNR):
- $TNR = \frac{TN + FP}{TN}$
- False Negative Rate (FNR):
- $FNR = \frac{TP + FN}{FN}$
- Precision:
- Precision =  $\frac{True\ Positive}{True\ Positive + False\ Positive}$
- Recall:
- Recall =  $\frac{True\ Positive}{True\ Positive + False\ Negative}$
- F1-score:
- $F1 = 2 * \frac{Recall * Precision}{Recall + Precision}$
- RMSE:
- $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$

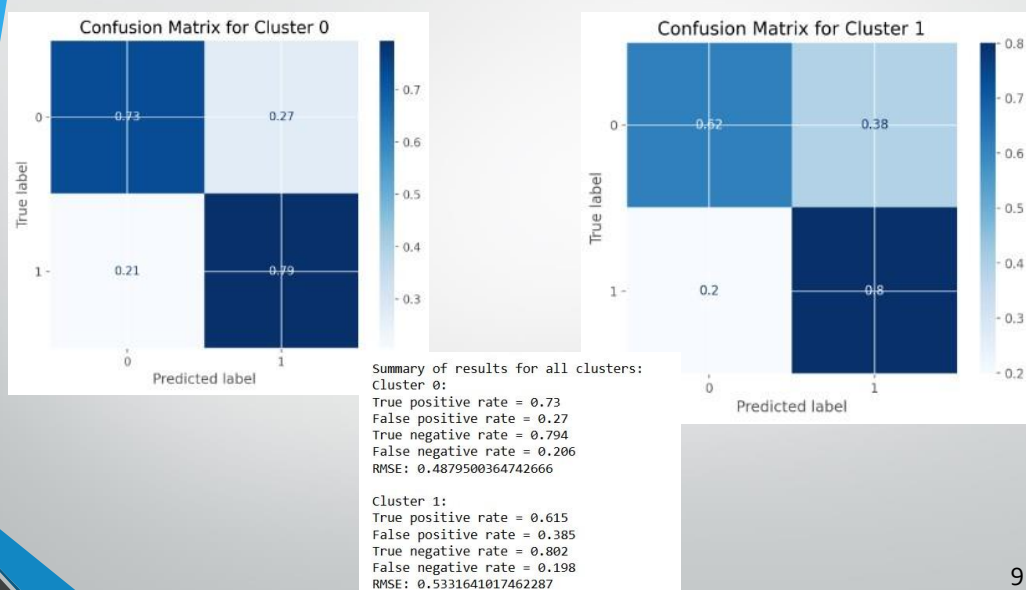
7

## Багатошарова нейронна мережа



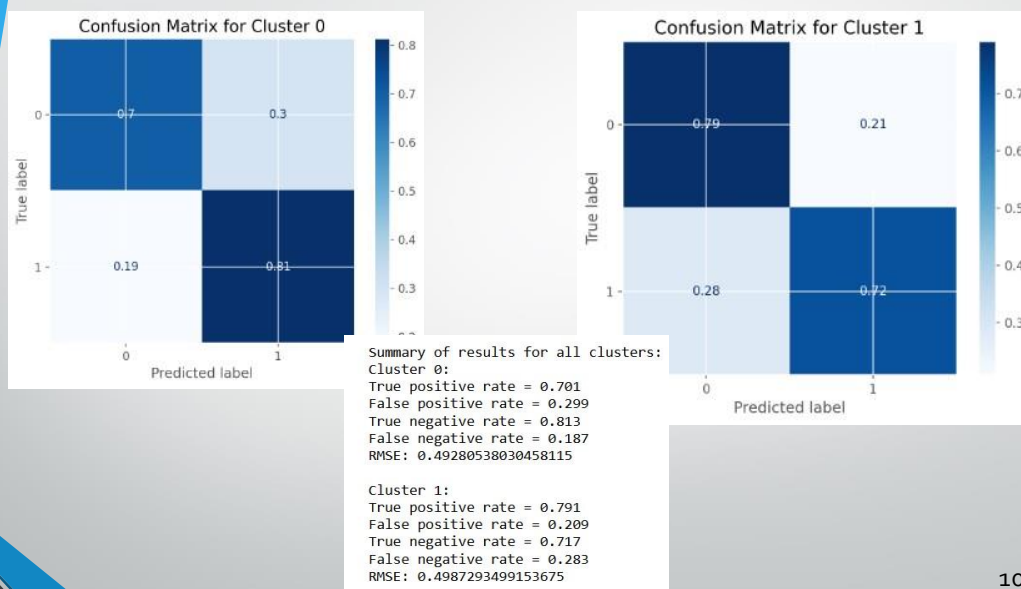
8

## Мережа довгої короточасної пам'яті (LSTM)



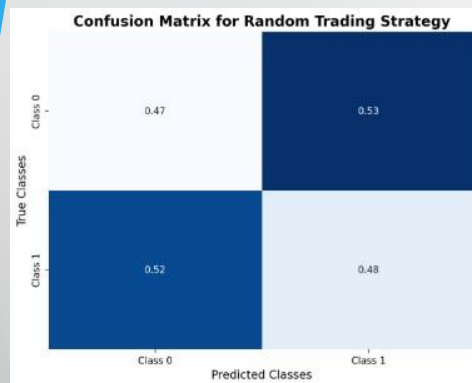
9

## Random Forest



10

## Рандомний торговий механізм



```
Random Trading Strategy - Classification Report:
              precision    recall  f1-score   support

     0       0.45     0.47     0.46     285
     1       0.50     0.48     0.49     315

 accuracy          0.47     600
 macro avg         0.47     0.47     0.47     600
 weighted avg     0.47     0.47     0.47     600
```

```
Random Trading Strategy - Confusion Matrix (normalized):
[[0.47017544 0.52982456]
 [0.52380952 0.47619048]]
Random Trading Strategy - RMSE: 0.725718035235908
```

Summary of results for Random Trading Strategy:

True positive rate = 0.47  
 False positive rate = 0.53  
 True negative rate = 0.476  
 False negative rate = 0.524  
 RMSE: 0.725718035235908

11

## Висновки


Метод	Номер кластера	Точність (Accuracy)	Precision (середнє)	Recall (середнє)	F1-score	RMSE
DNN	0	0,75	0,76	0,75	0,75	0,4976
	1	0,77	0,78	0,77	0,77	0,4832
LSTM	0	0,76	0,76	0,76	0,76	0,4879
	1	0,72	0,72	0,71	0,71	0,5332
Random Forest	0	0,76	0,76	0,76	0,76	0,4928
	1	0,75	0,75	0,75	0,75	0,4987
Рандомний торговий показник		0,47	0,47	0,47	0,47	0,7257

Згідно отриманих результатів DNN показує найкращі результати в кластері 1, де дані характеризуються нижчою волатильністю. Цей метод забезпечує високий рівень точності у виявленні позитивних класів, що відображається у найвищому показнику True Positive Rate, а також має найменшу середньоквадратичну похибку (RMSE).

У свою чергу LSTM, навпаки, є найкращим вибором для кластеру 0, що характеризується високою волатильністю. Ця модель показує найкращі результати за показником True Positive Rate у кластері 0, а також найменшу похибку RMSE. Однак у кластері 1 її ефективність знижується через вищий рівень хибнопозитивних передбачень, що обмежує її універсальність.

Random Forest показує стабільні, але середні результати в обох кластерах. Цей метод є універсальним, однак він поступається нейронним мережам у точності розпізнавання складних закономірностей. Він демонструє збалансовані показники у кластері 0, проте програє DNN у кластері 1 за точністю та узгодженістю передбачень.

12



Дякую за увагу