

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

«До захисту допущено»

Завідувач кафедри

Віталія КОЙБІЧУК

(підпис)

(Ім'я та ПРІЗВИЩЕ)

_____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр
(бакалавр / магістр)

зі спеціальності 051 «Економіка»,
(код та назва)

освітньо-професійної програми Економічна кібернетика та бізнес аналітика
(освітньо-професійної / освітньо-наукової) (назва програми)

на тему: Використання машинного навчання для прогнозування змін курсів акцій на фондовому ринку

Здобувача (ки) групи ЕК-01а Пейчева Івана Івановича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ (підпис)

Іван Пейчев

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцентка, д. е. н., Ганна Яровенко
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2024

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ
Завідувачка кафедри
к.е.н., доцентка
_____ Віталія КОЙБІЧУК
“ ___ ” _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ НА
ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
(спеціальність 051 Економіка «Економічна кібернетика та бізнес аналітика»)

студенту 4 курсу, групи ЕК-01а

Пейчеву Івану Івановичу

(прізвище, ім'я, по батькові студента)

1. Тема роботи Використання машинного навчання для прогнозування змін курсів акцій на фондовому ринку.

затверджена наказом по університету від «8» травня 2024 року № 0486-VI

2. Термін подання студентом закінченої роботи «1» червня 2024 року

3. Мета кваліфікаційної роботи використання та оцінка моделей машинного навчання для прогнозування курсів цін акцій на фондовому ринку.

4. Об'єкт дослідження історичні данні цін акцій.

5. Предмет дослідження аналітичний, статистичний та програмний інструментарій для прогнозування змін курсів акцій.

6. Кваліфікаційна робота виконується на матеріалах історичних цін акцій AAPL та MSFT.

7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1 Фактори впливу на цінову динаміку акцій та методи прогнозування «1» квітня 2024 року

(назва – термін подання)

У розділі 1 необхідно охарактеризувати роль фундаментального та технічного аналізу, вплив новин та суспільних подій на цінову динаміку акцій, а також розглянути машинне навчання як метод для прогнозування цін акцій.

(зміст конкретних завдань до розділу, які повинен виконати студент)

Розділ 2 Експлоративний аналіз даних «1» травня 2024 року

(назва – термін подання)

У розділі 2 необхідно розробити концептуальну модель прогнозування цін на акції, оглянути основні характеристики датасету, а також реалізувати обраний метод прогнозування цін на акції.

(зміст конкретних завдань до розділу, які має виконати студент)

Розділ 3 Опис результатів моделювання та їх верифікація «6» травня 2024 року

(назва – термін подання)

У розділі 3 необхідно описати результати моделювання та прогнозування цін акцій, а також провести верифікацію отриманих прогнозів для оцінки їхньої точності та надійності.

(зміст конкретних завдань до розділу, які повинен виконати студент)

8. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Яровенко Г.М., доцент	Яровенко Г.М 01.04.24	Пейчев І.І. 01.04.24
2	Яровенко Г.М., доцент	Яровенко Г.М 01.05.24	Пейчев І.І. 01.05.24
3	Яровенко Г.М., доцент	Яровенко Г.М 06.05.24	Пейчев І.І. 06.05.24

9. Дата видачі завдання: «1» квітня 2024 року

Керівник кваліфікаційної роботи _____

(підпис)

Г. М. Яровенко

(ініціали, прізвище)

Завдання до виконання одержав _____

(підпис)

І. І. Пейчев

(ініціали, прізвище)

РЕФЕРАТ

кваліфікаційної роботи бакалавра на тему «ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ЗМІН КУРСІВ АКЦІЙ НА ФОНДОВОМУ РИНКУ»

студента Пейчева Івана Івановича
(прізвище, ім'я, по батькові)

Актуальність теми, обраної для дослідження, визначається тим, що

Мета кваліфікаційної роботи полягає у використанні та оцінці моделей машинного навчання для прогнозування курсів цін акцій на фондовому ринку.

Об'єктом дослідження є історичні данні цін акцій.

Предметом дослідження є аналітичний, статистичний та програмний інструментарій для прогнозування змін курсів акцій.

Задачами дослідження є:

- 1) провести аналіз факторів впливу на динаміку курсів цін акцій;
- 2) здійснити аналіз існуючих підходів до прогнозування цін акцій з використанням машинного навчання;
- 3) визначити найбільш ефективні алгоритми машинного навчання для вирішення завдання прогнозування;
- 4) розробити методіку підготовки даних для навчання моделей;
- 5) реалізувати обрану модель машинного навчання;
- 6) спрогнозувати курси акцій;
- 7) провести верифікацію отриманих прогнозів.

Для досягнення поставленої мети та задач дослідження були використані такі методи дослідження: аналіз та інтеграція, узагальнення, обґрунтування, порівнювання та систематизація, за допомогою яких були зроблені загальні висновки, методи статистичного аналізу та машинного навчання.

Інформаційною базою кваліфікаційної роботи є історичні дані цін наступних акцій: AAPL, MSFT (263 спостереження та 2 змінних в кожному)

Основний науковий результат кваліфікаційної роботи полягає у такому: були створені моделі для перевірки основних характеристик вхідних даних, обрану підходящу модель машинного навчання, її подальше навчання що дає змогу отримати прогноз щодо зміни курсу акцій на майбутні періоди.

Одержані результати можуть бути використані для удосконалення інвестиційних стратегій як окремих інвесторів, так і фінансових установ.

Результати апробації основних положень кваліфікаційної роботи розглядалися на науковій онлайн конференції «Цифрові навички: виклики та можливості».

Ключові слова: ML, машинне навчання, акція, прогнозування.

Зміст кваліфікаційної роботи викладено на 63 сторінках. Список використаних джерел із 35 найменувань, розміщений на 3 сторінках. Робота містить 0 таблиць, 46 рисунків, а також 4 додатки, розміщених на 4 сторінках.

Рік виконання кваліфікаційної роботи – 2024 рік.

Рік захисту роботи – 2024 рік

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ФАКТОРИ ВПЛИВУ НА ЦІНОВУ ДИНАМІКУ АКЦІЙ ТА МЕТОДИ ПРОГНОЗУВАННЯ	9
1.1 Роль фундаментального та технічного аналізу	9
1.1.1 Фундаментальний аналіз	9
1.1.2 Технічний аналіз.....	10
1.2 Вплив новин та суспільних подій.....	13
1.2.1 Аналіз новинного фону.....	13
1.2.2 Аналіз настроїв інвесторів.....	14
1.3 Машинне навчання для прогнозування цін акцій.....	16
1.3.1 Типи алгоритмів машинного навчання	16
1.3.2 Алгоритми прогнозування часових рядів	18
1.3.3 Ризики та обмеження	21
РОЗДІЛ 2 ЕКСПЛОРАТИВНИЙ АНАЛІЗ ДАНИХ	23
2.1 Концептуальна модель прогнозування цін на акції.....	23
2.2 Огляд основних характеристик датасету	25
2.3 Реалізація обраного методу прогнозування цін на акції	34
РОЗДІЛ 3 ОПИС РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ ТА ЇХ ВЕРИФІКАЦІЯ.....	47
3.1 Опис результатів моделювання та прогнозування цін акцій.....	47
3.2 Верифікація прогнозів	49
ВИСНОВОКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ.....	57

ВСТУП

Сучасний фондовий ринок є складною системою, де ціни акцій формуються під впливом численних факторів, включаючи макроекономічні показники, корпоративні новини, геополітичні події та ринкові настрої. Прогнозування цін акцій стало важливим завданням для інвесторів, фінансових аналітиків та інших учасників ринку. Використання машинного навчання для цього завдання стає все більш актуальним через здатність цих методів обробляти великі обсяги даних та виявляти складні закономірності, які важко виявити традиційними методами аналізу. Застосування алгоритмів машинного навчання може значно підвищити точність прогнозів та забезпечити ефективніші інвестиційні стратегії, що є критично важливим в умовах високої волатильності та конкурентності сучасного ринку.

Фондові ринки є важливими індикаторами економічного стану країн і глобальної економіки загалом. Щоденний обсяг торгів на цих платформах сягає мільярдів доларів, а зміни в цінах акцій можуть мати значний вплив на інвестиційні портфелі, пенсійні фонди та фінансові інститути по всьому світу. В умовах глобалізації та високої інтеграції світових ринків, події на одному ринку можуть швидко поширюватися і впливати на інші. Це підвищує значущість точних прогнозів для мінімізації ризиків і максимізації прибутків. У зв'язку з цим, впровадження новітніх технологій, таких як машинне навчання, для аналізу та прогнозування ринкових тенденцій стає не тільки інноваційним підходом, але й необхідністю для ефективного управління фінансовими активами.

Метою кваліфікаційної роботи є використання та оцінка моделей машинного навчання для прогнозування курсів цін акцій на фондовому ринку.

Об'єктом дослідження є історичні данні цін акцій.

Предметом дослідження є аналітичний, статистичний та програмний інструментарій для прогнозування змін курсів акцій.

Для досягнення мети дослідження поставлено наступні завдання:

1. Провести аналіз факторів впливу на динаміку курсів цін акцій;

2. Здійснити аналіз існуючих підходів до прогнозування цін акцій з використанням машинного навчання;
3. Визначити найбільш ефективні алгоритми машинного навчання для вирішення завдання прогнозування;
4. Розробити методіку підготовки даних для навчання моделей;
5. Реалізувати обрану модель машинного навчання;
6. Зпрогнозувати курси акцій;
7. Провести верифікацію отриманих прогнозів.

У ході дослідження використовуються такі методи:

- Методи статистичного аналізу.
- Методи аналізу даних для підготовки та попередньої обробки інформації.
- Методи машинного навчання.

Інформаційною базою дослідження є історичні дані цін наступних акцій: AAPL, MSFT (263 спостереження та 2 змінних в кожному), отримані з відкритих джерел фінансово інформаційних платформ та баз даних, на основі яких проводився аналіз, реалізовувалась модель та відбувався прогноз. Для здійснення розрахунків та візуалізації результатів було використано інтерактивне хмарне середовище для роботи з кодом Google Colaboratory й мова програмування Python відповідно.

РОЗДІЛ 1 ФАКТОРИ ВПЛИВУ НА ЦІНОВУ ДИНАМІКУ АКЦІЙ ТА МЕТОДИ ПРОГНОЗУВАННЯ

1.1 Роль фундаментального та технічного аналізу

1.1.1 Фундаментальний аналіз

Фундаментальний аналіз – це метод оцінки вартості активу, що базується на аналізі економічних та фінансових факторів, які впливають на діяльність компанії та її перспективи. Метою фундаментального аналізу є визначення справжньої (внутрішньої) вартості акції та порівняння її з ринковою ціною, щоб зрозуміти, чи є акція переоціненою, недооціненою, або справедливо оціненою[1].

Вплив макроекономічних показників

Макроекономічні показники відображають загальний стан економіки та можуть суттєво впливати на прибутковість компаній і, відповідно, на ціни їхніх акцій.

До основних належать:

- ВВП (валовий внутрішній продукт): зростання ВВП зазвичай свідчить про сприятливі умови для бізнесу, підвищення споживчого попиту та інвестицій, що позитивно впливає на доходи компаній та ціни акцій.
- Інфляція: висока інфляція може призвести до зростання витрат компаній, зниження купівельної спроможності споживачів та підвищення процентних ставок, що має негативний вплив.
- Процентні ставки: зміна процентних ставок впливає на вартість позикових коштів для компаній та привабливість інвестицій. Підвищення ставок зазвичай призводить до зниження цін акцій, оскільки інвестори шукають більш прибуткові альтернативи[2].

Вплив галузевих факторів

Специфіка галузі, в якій працює компанія, також має значний вплив на її успіх.

- Конкуренція: високий рівень конкуренції може обмежувати можливості компанії щодо підвищення цін та збільшення частки ринку, що негативно впливає на прибутковість та ціну акцій.

- Регулювання: зміни в законодавстві та державному регулюванні можуть створювати нові можливості або обмеження для компаній, що впливає на їхню діяльність.
- Технологічні зміни: швидкий розвиток технологій може призвести до появи нових продуктів та послуг, що змінюють конкурентний ландшафт та впливають на успіх компаній у довгостроковій перспективі[3].

Аналіз фінансового стану компанії

Фінансовий стан компанії є одним з ключових факторів, що визначають її інвестиційну привабливість та ціну акцій.

- Дохідність: аналіз доходів та прибутковості компанії дозволяє оцінити її ефективність та потенціал для зростання.
- Боргове навантаження: високий рівень боргу може обмежувати можливості компанії щодо інвестування та розвитку, підвищувати ризики та негативно впливати.
- Перспективи зростання: аналіз перспектив зростання компанії дозволяє оцінити її потенціал для збільшення доходів та прибутків у майбутньому, що позитивно впливає[4].

Використовуючи фундаментальний аналіз, інвестори можуть приймати більш обґрунтовані рішення щодо інвестування в акції, формувати ефективні інвестиційні стратегії. Цей підхід дозволяє глибше зрозуміти економічні, фінансові та інші якісні фактори, які впливають на вартість акцій, що в свою чергу допомагає знизити ризики та підвищити ймовірність отримання стабільного доходу. Таким чином, цей підхід є незамінним інструментом для інвесторів, які прагнуть до довгострокового успіху на фондовому ринку.

1.1.2 Технічний аналіз

Технічний аналіз – це методика прогнозування майбутніх цін на акції, що базується на вивченні історичних даних про ціни, обсяги торгів та інші ринкові показники. На відміну від фундаментального аналізу, який зосереджується на внутрішній вартості акцій, технічний аналіз припускає, що вся необхідна інформація вже відображена в ціні та обсязі торгів[5].

Використання технічних індикаторів

Технічні індикатори – це математичні формули, що застосовуються до історичних даних про ціни та обсяги торгів, щоб виявити певні закономірності та тенденції на ринку.

До найпоширеніших індикаторів належать:

- Середні ковзні (Moving Averages): відображають середню ціну акції за певний період часу, допомагаючи визначити напрямок тренду та потенційні точки розвороту.
- Індекси сили (Relative Strength Index, RSI): вимірюють силу та швидкість зміни ціни, допомагаючи визначити перекуплені та перепродані зони.
- Об'єми торгів (Volume): показують кількість акцій, що були продані та куплені за певний період, допомагаючи оцінити силу тренду та інтерес інвесторів[6].

Визначення трендів та патернів

Даний аналіз приділяє велику увагу виявленню трендів (зростаючих, спадних або бокових) та патернів (графічних фігур), які можуть вказувати на майбутні зміни ціни.

Тренди: визначаються за допомогою ліній тренду, середніх ковзних та інших індикаторів.

Патерни: включають різноманітні фігури, такі як "голова та плечі", "подвійне дно" та "прапор", що можуть сигналізувати про потенційні розвороти тренду або його продовження[7].

Деякі методи технічного аналізу базуються на специфічних теоріях, таких як:

- Теорія Доу: стверджує, що ринок рухається в рамках трендів, які складаються з трьох фаз: накопичення, зростання та розподілу[8].

— Хвильовий аналіз: припускає, що цінові рухи на ринку відбуваються у вигляді повторюваних хвильових патернів, що дозволяє прогнозувати майбутні зміни ціни[9].

Варто зауважити, що аналіз має свої обмеження, які включають:

1. Залежність від історичних даних: аналіз базується на минулих цінових рухах та обсягах торгів, що не завжди є надійним показником майбутніх змін.
2. Суб'єктивність інтерпретації: різні аналітики можуть по-різному тлумачити одні й ті самі графіки та індикатори, що може призводити до різних висновків і рекомендацій.
3. Ризик помилкових сигналів: індикатори технічного аналізу можуть генерувати хибні сигнали, що може призвести до невдалих інвестиційних рішень.
4. Відсутність врахування фундаментальних факторів: технічний аналіз не враховує фінансовий стан компаній, макроекономічні показники та інші фундаментальні чинники, які можуть впливати на ринкові ціни.
5. Ринковий шум: короткострокові коливання ринку, викликані випадковими подіями або спекуляціями, можуть ускладнювати точний прогноз за допомогою технічного аналізу[10].

Технічний аналіз є корисним інструментом для короткострокового прогнозування, але його ефективність значною мірою залежить від досвіду та вміння аналітика інтерпретувати дані та використовувати різні індикатори та методи. Правильне застосування технічного аналізу дозволяє виявляти тренди, визначати оптимальні точки входу та виходу з ринку, а також розпізнавати потенційні сигнали змін. Згаданий підхід не гарантує успіху і має свої обмеження, тому його варто використовувати в поєднанні з іншими методами аналізу для отримання більш повної картини ринкових умов.

1.2 Вплив новин та суспільних подій

1.2.1 Аналіз новинного фону

Інформаційний фон, що включає в себе новини, суспільні події та настрої інвесторів, може суттєво впливати на цінову динаміку акцій, викликаючи як короткострокові коливання, так і довгострокові тренди[11].

Вплив корпоративних новин:

- Фінансові звіти: публікація квартальних та річних звітів, що відображають фінансові результати компанії, може призвести до різких змін ціни акцій, залежно від того, чи відповідають ці результати очікуванням ринку.
- Злиття та поглинання: оголошення про злиття або поглинання може впливати на ціну акцій обох компаній, як позитивно, так і негативно, залежно від очікуваних синергій та ризиків.
- Зміни в керівництві: зміна генерального директора або інших ключових керівників може викликати зміни в стратегії компанії та вплинути на довіру інвесторів, що відбивається на ціні акцій.

Синергія – це явище, при якому спільна дія двох або більше елементів дає кращий результат, ніж сума їх індивідуальних дій. Іншими словами, синергія виникає, коли співпраця між різними компаніями або людьми призводить до більшої ефективності, продуктивності або успіху, ніж вони могли б досягти окремо.

Вплив політичних та економічних подій:

- Вибори, зміна уряду: політична нестабільність або зміни в економічній політиці можуть створювати невизначеність та впливати на загальні настрої, що відбивається на активах.
- Торговельні війни, санкції: обмеження міжнародної торгівлі та економічні санкції можуть нести негативний вплив на компанії, що залежать від експорту або імпорту, що призводить до зниження цін акцій.

— Зміни в монетарній політиці: рішення центральних банків щодо процентних ставок та інших інструментів монетарної політики можуть впливати на вартість капіталу та привабливість інвестицій[12].

Вплив природних катастроф та інших форс-мажорних обставин:

— Стихійні лиха, техногенні катастрофи: природні явища та інші форс-мажорні обставини можуть призвести до зупинки виробництва, пошкодження майна та інших фінансових втрат для компаній, що в подальшому негативно впливає.

— Епідемії, пандемії: глобальні ситуації можуть впливати на економічну активність, споживчий попит та ланцюжки постачання, що призводить до зниження вартості цінних паперів в багатьох галузях[13].

Аналіз новинного фону та розуміння його впливу на цінову динаміку акцій дозволяє інвесторам приймати більш обґрунтовані рішення та адаптувати свої інвестиційні стратегії до мінливих умов ринку.

1.2.2 Аналіз настроїв інвесторів

Аналіз настроїв інвесторів – це процес оцінки емоцій та настроїв учасників ринку щодо конкретних акцій, галузей чи ринку в цілому. Розуміння настроїв інвесторів допомагає передбачити потенційні зміни в попиті та пропозиції, а отже, і цінові рухи[14].

Інструменти та методи аналізу

I. Аналіз соціальних мереж та медіа:

— Моніторинг соціальних мереж: відстеження згадок про компанії та акції в соціальних мережах, таких як Twitter, Facebook, Reddit, тощо, дозволяє виявити загальний настрій інвесторів та виявити потенційні тригери для змін цін.

— Аналіз новинних статей та блогів: використання інструментів обробки природної мови для аналізу тональності новинних статей та блогів, що стосуються компаній та ринку в цілому, допомагає оцінити рівень оптимізму чи песимізму[15].

II. Опитування:

- Проведення опитувань серед індивідуальних та інституційних інвесторів дозволяє отримати пряму інформацію про їхні очікування та настрої.
- Аналіз звітів інвестиційних компаній та аналітиків: звіти від провідних інвестиційних компаній та аналітиків часто містять інформацію про очікувані тренди та настрої інвесторів.

III. Визначення рівня оптимізму/песимізму на ринку

- Індекси настроїв: існують різні індекси, що агрегують дані з різних джерел для оцінки загального настрою інвесторів, наприклад, Індекс страху та жадібності (Fear & Greed Index)[16].
- Аналіз обсягів торгів: збільшення обсягів торгів може свідчити про підвищений інтерес, як позитивний, так і негативний, що допомагає оцінити силу тренду.
- Аналіз опціонного ринку: співвідношення Put/Call – це показник, що відображає співвідношення опціонів на продаж (Put) до опціонів на купівлю (Call) і може свідчити про переважання песимістичних або оптимістичних настроїв[17].

Роль машинного навчання

Машинне навчання відіграє важливу роль у аналізі настроїв в соціальних мережах, дозволяючи автоматично обробляти та інтерпретувати величезні обсяги текстових даних, створених користувачами. Завдяки методам обробки природної мови (NLP), алгоритми можуть визначати тональність повідомлень, розпізнавати позитивні, негативні або нейтральні настрої та виявляти ключові теми обговорень. Це надає компаніям можливість швидко реагувати на зміни в настроях споживачів, покращувати клієнтське обслуговування, а також розробляти ефективні маркетингові

стратегії. Водночас, аналіз настроїв допомагає виявляти потенційні кризи та проблеми, що дозволяє приймати практичні заходи для їх вирішення[18].

Таким чином, використання машинного навчання для аналізу настроїв у соціальних мережах стає важливим інструментом для розуміння громадської думки та прийняття обґрунтованих бізнес-рішень.

Сентимент-аналіз є корисним інструментом для інвесторів, але його слід використовувати з обережністю та в поєднанні з іншими методами аналізу, оскільки настрої можуть бути мінливими та не завжди відображають фундаментальні фактори.

1.3 Машинне навчання для прогнозування цін акцій

1.3.1 Типи алгоритмів машинного навчання

Машинне навчання (ML) є ключовою складовою сучасних інформаційних технологій, яка дозволяє системам автоматично вчитися та покращуватися з досвідом без явного програмування. Це підгалузь штучного інтелекту, яка базується на ідеї, що машини можуть навчатися з даних, розпізнавати шаблони і приймати рішення з мінімальним втручанням людини. Завдяки розвитку обчислювальних потужностей і доступності великих обсягів даних, ML набуло значної популярності і застосовується в різних сферах: від медицини до фінансів, від обробки зображень до автономного транспорту.

Завдяки своїй здатності аналізувати величезні обсяги даних та виявляти складні залежності, машинне навчання стає все більш популярним інструментом для прогнозування. Алгоритми можуть враховувати різноманітні фактори, такі як історичні ціни, фінансові показники компаній, макроекономічні дані, настрої інвесторів, та будувати моделі, що прогнозують майбутню динаміку цін[19].

Розуміння різних типів алгоритмів машинного навчання є критично важливим для розробки ефективних і надійних систем. Методи можна класифікувати за

кількома критеріями, серед яких найбільш значущими є методи навчання та типи задач.

Методи навчання:

- Навчання з учителем (Supervised Learning): цей метод передбачає наявність міток у навчальному наборі даних. Алгоритми вчать на основі зразків, де кожному входу відповідає певний вихід. Приклади включають регресію та класифікацію.
- Навчання без учителя (Unsupervised Learning): цей метод не передбачає міток у навчальному наборі даних. Алгоритми намагаються знайти приховані структури в даних. Типові приклади включають кластеризацію та асоціаційні правила.
- Напівконтрольоване навчання (Semi-Supervised Learning): поєднує елементи як навчання з учителем, так і без учителя, використовуючи невелику кількість мічених даних разом з великою кількістю немічених.
- Навчання з підкріпленням (Reinforcement Learning): алгоритми цього типу вчать через взаємодію з середовищем, отримуючи винагороди або покарання за свої дії[20].

Типи задач:

- Класифікація: завдання розподілу об'єктів по категоріях. Приклади включають розпізнавання зображень та діагностику хвороб.
- Регресія: завдання прогнозування безперервних значень. Використовується, наприклад, для передбачення цін на акції або погоди.
- Кластеризація: завдання групування об'єктів за схожістю без попередньо визначених міток. Використовується для сегментації ринку або виявлення аномалій.
- Зниження розмірності: завдання зменшення кількості змінних, використовуючи їх головні компоненти. Використовується для візуалізації даних та підготовки їх до інших алгоритмів.

Розуміння цих типів алгоритмів дозволяє вибрати найефективніші підходи для вирішення конкретних задач, оптимізуючи процес навчання та підвищуючи точність і надійність прогнозів.

До основних переваг використання машинного навчання належать:

1. Здатність аналізувати великі обсяги даних: алгоритми ML можуть ефективно обробляти та аналізувати величезні масиви інформації, що недоступно для традиційних методів аналізу.
2. Виявлення складних залежностей: ML моделі можуть виявляти нелінійні та взаємопов'язані фактори, що впливають на активи, забезпечуючи більш точні прогнози.
3. Автоматизація процесу: після навчання моделі ML можуть автоматично робити прогнози на основі нових даних, що економить час та зусилля[21].

1.3.2 Алгоритми прогнозування часових рядів

Часові ряди – це послідовності даних, зібраних через рівні проміжки часу, наприклад, щоденні ціни акцій. Алгоритми часових рядів спеціально розроблені для аналізу таких даних та виявлення закономірностей, що дозволяє прогнозувати майбутні значення.

До основних алгоритмів прогнозування часових рядів:

- I. Прості методи:
 - Ковзне середнє (Moving Average): використовується для згладжування короткострокових коливань та виявлення основних тенденцій.
 - Експоненційне згладжування (Exponential Smoothing): метод, що надає більшу вагу останнім спостереженням для швидшого реагування на зміни[22].
- II. Статистичні методи:

- ARIMA (Autoregressive Integrated Moving Average): комбінує авторегресію, інтеграцію та ковзне середнє для створення потужного інструменту прогнозування.
- SARIMA (Seasonal ARIMA): розширення ARIMA, яке враховує сезонність у даних.

III. Машинне навчання:

- Лінійна регресія: проста модель, що передбачає лінійну залежність між вхідними змінними та прогнозованими значеннями.
- Support Vector Machines (SVM): використовується для знаходження оптимальних гіперплощин, що розділяють дані.
- Дерева рішень та випадкові ліси (Decision Trees and Random Forests): моделі, що можуть враховувати нелінійні залежності.
- Нейронні мережі: здатні виявляти складні шаблони та взаємозв'язки у даних.

IV. Глибоке навчання:

- Recurrent Neural Networks (RNN): особливо ефективні для обробки послідовних даних завдяки своїй архітектурі зворотних зв'язків.
- Long Short-Term Memory (LSTM): різновид RNN, що добре справляється з проблемою зникнення градієнтів та зберігає довгострокові залежності.
- Transformers: сучасні моделі, що використовуються для прогнозування часових рядів завдяки своїй здатності обробляти довгі послідовності даних паралельно.

V. Гібридні моделі:

- Поєднання статистичних методів та машинного навчання. Наприклад, використання ARIMA для попередньої обробки даних перед застосуванням нейронних мереж.
- Мультимодальні підходи: використання кількох моделей одночасно для підвищення точності прогнозування шляхом об'єднання їхніх результатів[23].

Для нестационарних рядів може знадобитись попередня обробка даних, наприклад, декомпозиція.

Декомпозиція – це процес перетворення нестационарного часового ряду на стаціонарний шляхом усунення сезонності та тренду. Важливо зазначити, що декомпозицію необхідно виконувати лише для деяких моделей, наприклад, регресійних моделей, таких як ARIMA/SARIMA.

RNN моделі можуть страждати від проблеми "зникаючого градієнта", що ускладнює навчання на довгих послідовностях. LSTM моделі розв'язують цю проблему завдяки своїй особливій архітектурі.

Градієнт – це вектор, який містить часткові похідні функції помилки (втрат) щодо кожного параметра (ваги) моделі. Він показує напрямок та величину змін, необхідних для зменшення функції помилки. Градієнт використовується в алгоритмі зворотного поширення помилки (backpropagation) для оновлення ваг мережі. Це здійснюється за допомогою оптимізаційних алгоритмів, таких як градієнтний спуск, які регулюють ваги, щоб мінімізувати помилку мережі.

Ваги – це параметри нейронної мережі, які визначають силу зв'язків між нейронами. Кожен нейрон у мережі множить свій вхідний сигнал на відповідну вагу, а потім передає результуюче значення через активаційну функцію. В процесі навчання ваги оновлюються на основі градієнтів, щоб нейронна мережа могла краще узгоджувати свої виходи з правильними відповідями[24].

Разом, градієнти і ваги формують основу навчання нейронних мереж. Градієнти вказують, як саме потрібно скорегувати ваги, щоб зменшити помилку, а ваги визначають, як саме нейрони перетворюють вхідні дані на вихідні.

1.3.3 Ризики та обмеження

Попри свою перспективність, машинне навчання для прогнозування не позбавлене ризиків та обмежень, які важливо розуміти для ефективного використання.

До ключових викликів та обмежень належить:

Перенавчання, недонавчання та упередженість моделей.

Перенавчання (overfitting) виникає, коли ML модель занадто добре "запам'ятовує" тренувальні дані, але погано узагальнює на нових даних. Це призводить до зниження точності прогнозування та неадекватних результатів.

Недонавчання (underfitting) виникає, коли ML модель не здатна захопити основні залежності в даних, що призводить до поганої продуктивності як на тренувальних, так і на тестових даних. Це може відбуватися через надто просту модель, недостатню кількість характеристик або через недостатню кількість тренувальних даних. Недонавчання також призводить до низької точності прогнозування та неспроможності моделі адекватно відображати реальні залежності.

Упередженість (bias) виникає, коли ML модель має систематичну помилку в прогнозуванні, наприклад, постійно занижує або завищує ціни. Упередженість може виникати через неякісні дані, неправильний вибір алгоритму або помилки в налаштуванні моделі.

Складність інтерпретації результатів.

ML моделі, особливо складні, такі як нейронні мережі, часто працюють як "чорний ящик", тобто важко зрозуміти, як вони прийшли до своїх прогнозів, який ускладнює інтерпретацію результатів та розуміння причин змін, що може знижувати довіру до моделей.

Необхідність великого обсягу даних та обчислювальних ресурсів.

Зазвичай для ефективного навчання ML моделей необхідні великі обсяги якісних даних, що не завжди доступні, особливо для нових компаній або малоліквідних акцій.

Малоліквідні цінні папери – ті, якими здійснюються небагато операцій на фондовому ринку, і які характеризуються низьким обсягом торгів.

Навчання складних моделей, таких як LSTM, може потребувати значних обчислювальних ресурсів, що підвищує вартість розробки та використання ML моделей.

Як мінімізувати ризики

1. Важливо ретельно валідувати та тестувати ML моделі на нових даних, щоб оцінити їхню точність та уникнути перенавчання.
2. Використання різних алгоритмів: порівняння результатів різних алгоритмів допомагає виявити найбільш ефективний та надійний метод для конкретного завдання.
3. Інтерпретація моделей: існують методи інтерпретації ML моделей, такі як LIME та SHAP, що дозволяють зрозуміти, які фактори найбільше впливають на прогнози.
4. Оптимізація обчислювальних ресурсів: використання хмарних платформ та спеціалізованого обладнання може знизити витрати на навчання ML моделей[25].

Розуміння ризиків та обмежень машинного навчання дозволяє використовувати ML моделі більш ефективно та приймати більш обґрунтовані рішення.

РОЗДІЛ 2 ЕКСПЛОРАТИВНИЙ АНАЛІЗ ДАНИХ

2.1 Концептуальна модель прогнозування цін на акції

Основна мета побудови концептуальної моделі для аналізу та прогнозування курсу акцій полягає у визначенні ключових факторів, що впливають на прогноз курсу акцій, з більшою точністю. Ця модель необхідна для прийняття обґрунтованих інвестиційних рішень і для розробки стратегій пом'якшення фінансових ризиків.

Постановка проблеми:

Прогнозування майбутніх цін на акції, виявлення недооцінених акцій або оцінку впливу ринкових подій на ціни акцій.

Джерела даних:

Модель спирається на статистичні дані з джерела-фінансові бази даних Yahoo Finance[26], Це джерело даних забезпечує необхідні дані для точного моделювання та прогнозування.

Вибір аналітичних інструментів:

Для реалізації концептуальної моделі була обрана така мова програмування, як Python, завдяки її обширним бібліотекам для аналізу даних, статистичної оцінки та машинного навчання. Такі бібліотеки, як Pandas, NumPy і TensorFlow, сприяють розробці складних моделей для прогнозування курсу акцій.

Концептуальна модель прогнозування цін на акції складається з наступних етапів, представлених на рисунку 2.1.1:

Вхідні змінні моделі (X, Y):

На цьому етапі визначаються всі змінні, які будуть аналізуватися. Це можуть бути фінансові індикатори, економічні показники, новини, історичні дані про ціну акцій та інше.

Огляд основних характеристик:

Після визначення вхідних змінних проводиться їхній аналіз та обробка. На цьому етапі здійснюється нормалізація даних, виявлення кореляцій між змінними, усунення можливих аномалій та пропусків у даних.

Машинне навчання. Нейронна мережа:

Використовуючи оброблені дані, будується модель на основі машинного навчання. Нейронна мережа обирається як метод прогнозування завдяки її здатності обробляти великі обсяги даних і виявляти складні залежності між змінними. Процес навчання включає в себе налаштування гіперпараметрів моделі, визначення архітектури нейронної мережі та навчання на тренувальних даних.

Результат. Отриманий прогноз:

Після завершення навчання моделі здійснюється прогнозування цін на акції на основі вхідних даних. Результатом є прогнозні значення цін, які будуть використані для подальшого аналізу.

Перевірка побудованої моделі на точність:

На заключному етапі проводиться валідація моделі для оцінки її точності та надійності. Зазвичай це робиться шляхом порівняння прогнозних значень з фактичними даними. Якщо точність моделі є недостатньою, проводяться додаткові налаштування та повторне навчання.



Рисунок 2.1.1 – Концептуальна модель прогнозування цін на акції

Таким чином, концептуальна модель прогнозування цін на акції включає підбір та обробку вхідних даних, побудову та навчання нейронної мережі, отримання прогнозних результатів та перевірку моделі на точність.

2.2 Огляд основних характеристик датасету

Для подальшого аналізу було взято тижневі дані за останні 5 років акцій AAPL(див. додаток Б) та MSFT(див. додаток В), дві змінні кожної акції, Date та Adjusted Close замість Close цін акцій, оскільки Adjusted Close враховує корпоративні події (дивіденди, спліти, злиття), що впливають на вартість акцій, забезпечуючи точніші історичні дані та полегшуючи технічний аналіз. Це дозволяє отримати реалістичнішу оцінку вартості активів, сприяє об'єктивному порівнянню різних компаній та забезпечує більш достовірні дані для побудови прогнозів і прийняття інвестиційних рішень.

Здійснення аналізу часового ряду:

Імпортовано необхідні бібліотеки для аналізу даних та часових рядів. Бібліотека pandas[27] використовується для роботи з даними, matplotlib[28] та seaborn[29] для візуалізації, statsmodels[30] для аналізу та моделювання часових рядів: pandas забезпечує роботу з табличними даними, matplotlib.pyplot основна бібліотека для побудови графіків, seaborn бібліотека для побудови статистичних графіків, seasonal_decompose функція для декомпозиції часових рядів на тренд, сезонність та залишкові компоненти, adfuller тест на одиничний корінь для перевірки стаціонарності часового ряду, plot_acf функція для побудови автокореляційної функції часового ряду (рис. 2.2.1).

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf
```

Рисунок 2.2.1 – Імпортовано необхідні бібліотеки.

Завантажено дані з файлу у DataFrame df. Змінено назву стовпця #Adj Close на Adj Close. Стовпець Date перетворено у формат дати та встановлено як індекс DataFrame (рис. 2.2.2). Виведено останні 10 рядків кожного набору даних для перегляду (рис. 2.2.3).

```
df=pd.read_csv("AAPL_filtered.csv")
date=df["Date"]
df.rename(columns={'#Adj Close': 'Adj Close'}, inplace=True)
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')
df.set_index('Date', inplace=True)
print(df.tail(10))
```

Рисунок 2.2.2 – Завантаження даних, налаштування форматів

Date	Adj Close	Date	Adj Close
2024-03-25	171.247726	2024-03-18	427.968048
2024-04-01	169.350296	2024-03-25	419.962494
2024-04-08	176.310867	2024-04-01	424.753845
2024-04-15	164.776505	2024-04-08	421.140381
2024-04-22	169.070679	2024-04-15	398.401398
2024-04-29	183.131607	2024-04-22	405.588440
2024-05-06	182.802063	2024-04-29	405.927826
2024-05-13	189.869995	2024-05-06	413.993256
2024-05-20	186.880005	2024-05-13	419.453400
2024-05-23	186.880005	2024-05-20	429.040009

Рисунок 2.2.3 – Виведені перші 10 рядків набору даних, ліворуч AAPL, праворуч MSFT.

Визначення розподілу

Побудовано гістограму для стовпця Adj Close (рис. 2.2.4). Гістограма показує розподіл значень скоригованої ціни закриття акцій, що дозволяє візуально оцінити їх розподіл та частотність.

```
df.hist('Adj Close')
```

Рисунок 2.2.4 – Побудова гістограми.

Результати побудови гістограм для акцій (рис. 2.2.5). Спостерігається нерівномірність, розподіл відмінний від нормального.

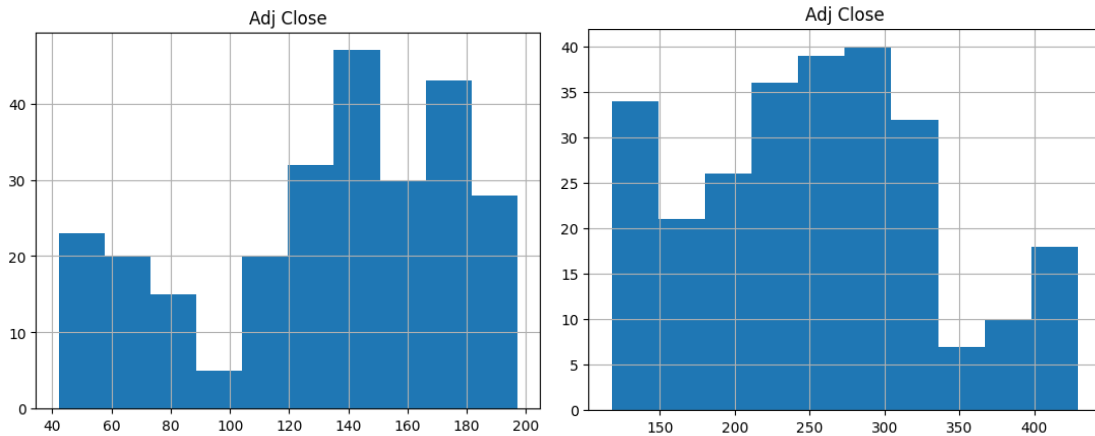


Рисунок 2.2.5 – Гістограми розподілу, ліворуч AAPL, праворуч MSFT.

Викиди

Побудовано boxplot для стовпця Adj Close, що дозволяє візуально оцінити розподіл значень, визначити медіану, квартилі та наявність викидів (рис. 2.2.6).

Вигляд boxplot підтверджує, що дані не мають викидів (рис. 2.2.7).

```
plt.figure(figsize=(5, 10))
sns.boxplot(y=df['Adj Close'])
plt.xlabel('Adjusted Close Price')
plt.title('Boxplot of Adjusted Close Prices')
plt.show()
```

Рисунок 2.2.6 – Побудова boxplot для стовпця Adj Close.

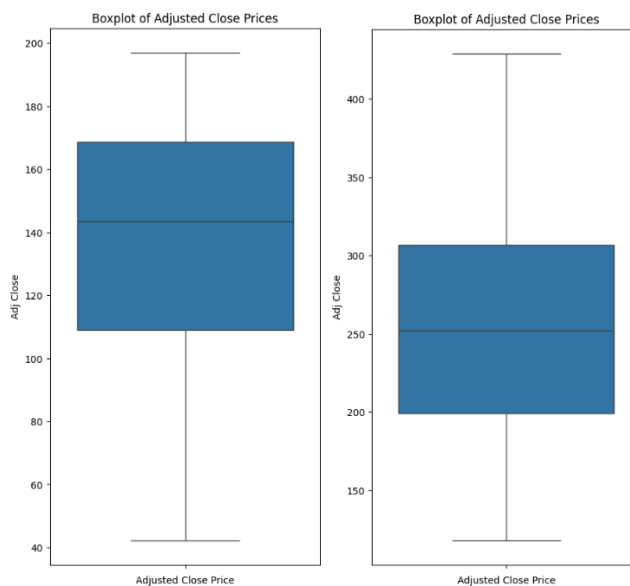


Рисунок 2.2.7 – Boxplot, ліворуч AAPL, праворуч MSFT.

Перевірка на NaN та основні характеристики

Використано метод `describe()` для отримання короткої статистики, основних характеристик числових ознак. Додатково підраховано кількість відсутніх значень (NaN) у кожному стовпці DataFrame за допомогою методу `isna().sum()` (рис. 2.2.8).

```
print(df.describe())
print("NaN values:",df.isna().sum())
```

Рисунок 2.2.8 – Метод для перевірки на NaN та отримання статистики

На основі отриманих статистичних описів зміненої ціни закриття Adj Close для двох наборів даних можна зробити наступні висновки.

Перший набір даних містить 263 спостереження з середнім значенням 132.12, стандартним відхиленням 43.44, мінімальним значенням 42.24 та максимальним значенням 197.05. Квартилі цього набору становлять 109.05 (25%), 143.41 (50%), та 168.66 (75%).

Другий набір даних також містить 263 спостереження, але з вищим середнім значенням 254.33 і більшим стандартним відхиленням 79.24. Мінімальне значення другого набору становить 117.95, а максимальне – 429.04. Квартилі цього набору

складають 199.34 (25%), 251.86 (50%), та 306.58 (75%). Це свідчить про те, що другий набір даних має більшу варіативність і вищі значення Adj Close.

NAN значення відсутні в обох наборах даних.

count	263.000000	count	263.000000
mean	132.116168	mean	254.334240
std	43.441794	std	79.244816
min	42.239899	min	117.953087
25%	109.046123	25%	199.336678
50%	143.408905	50%	251.857788
75%	168.656318	75%	306.584946
max	197.050949	max	429.040009
NAN values: Adj Close	0	NAN values: Adj Close	0
dtype: int64		dtype: int64	

Рисунок 2.2.9 – , ліворуч AAPL, праворуч MSFT.

Побудовано графіки для відображення змін скоригованої ціни закриття акцій з плином часу (рис. 2.2.10 – 2.2.12).

```
# Побудова графіка
df['Adj Close'].plot(figsize=(14,6))
plt.xlabel('Date')
plt.ylabel('Adjusted Close Price')
plt.title('Adjusted Close Price Over Time')
plt.grid(True)

# Відображення графіка
plt.show()
```

Рисунок 2.2.10 – Метод для побудови графіка.



Рисунок 2.2.11 – Графік курсу AAPL.

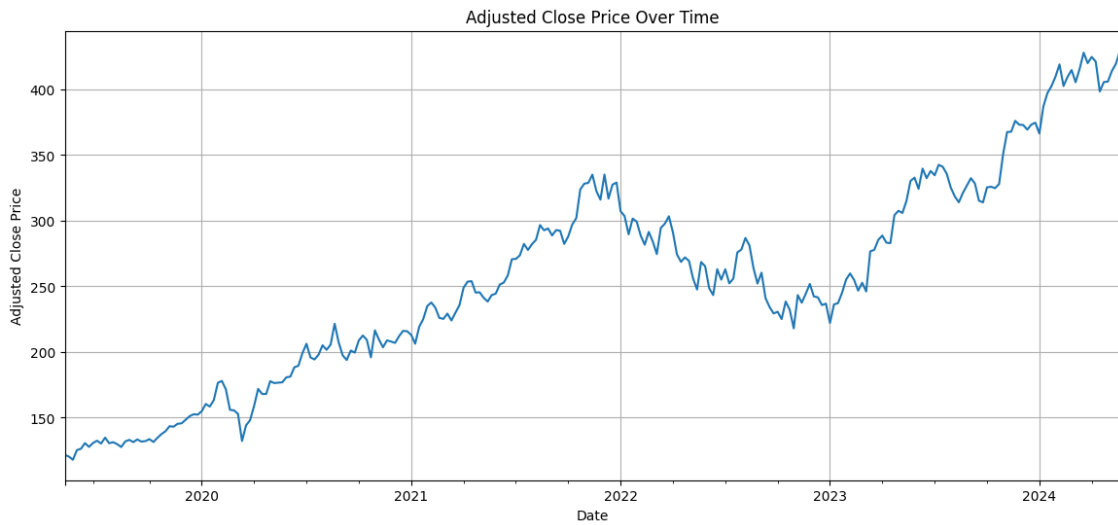


Рисунок 2.2.12 – Графік курсу MSFT.

Сезонність, тренд та шум

Виконано декомпозицію часового ряду скоригованої ціни закриття акцій на тренд, сезонність та залишкові компоненти за допомогою функції `seasonal_decompose` з періодом 7 днів. Побудовано графіки для кожного компонента(рис. 2.2.13).

На графіках обох акцій спостерігається тренд на зростання, також наявність шуму (рис. 2.2.14 та рис. 2.2.15).

```
decomposed = seasonal_decompose(df['Adj Close'], period=7)
decomposed.plot()
```

Рисунок 2.2.13 – Виконання методу.

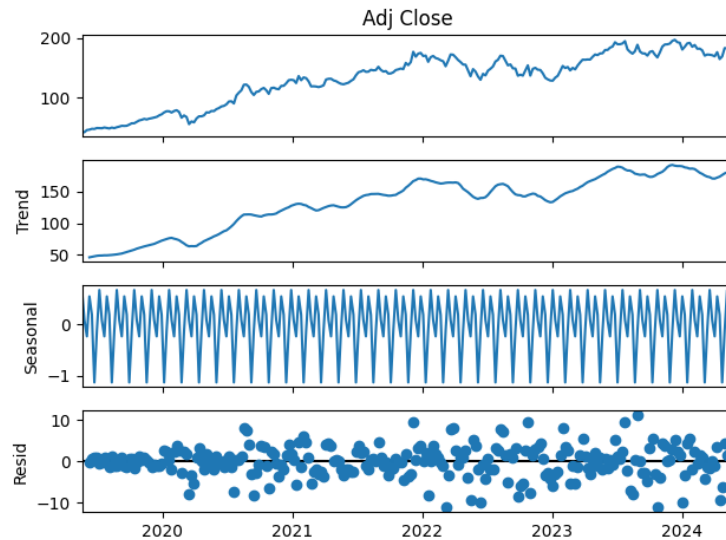


Рисунок 2.2.14 – Тренд, сезонність та залишкові компоненти для AAPL.

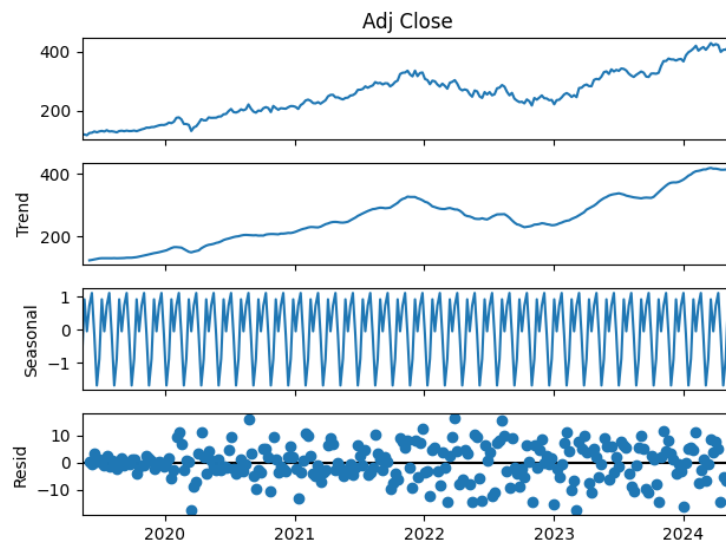


Рисунок 2.2.15 – Тренд, сезонність та залишкові компоненти для MSFT.

Доповнений тест Дики-Фулера (ADF)

Для перевірки стаціонарності часового ряду було створено дві функції: `adfuller_test` та `stat_check` (рис. 2.2.16).

Функція `adfuller_test` виконує тест Дики-Фулера для перевірки стаціонарності часового ряду. Вона повертає р-значення, статистику тесту та критичні значення.

Функція `stat_check` обчислює ковзне середнє та ковзне стандартне відхилення з вікном 12. Далі вона візуалізує оригінальний ряд, ковзне середнє та ковзне стандартне відхилення на графіку.

Після цього виконується тест Дікі-Фуллера, і на основі р-значення приймається рішення про стаціонарність ряду. Якщо р-значення менше заданого порогу (0.05 за замовчуванням), ряд вважається стаціонарним.

Функція `stat_check` застосована до часового ряду `Adj Close`:

```
def adfuller_test(df):
    adf=adfuller(df)
    statistic= adf[0]
    p_value=adf[1]
    critical_values=adf[4]
    return p_value,statistic,critical_values
def stat_check(df,p_value_check=0.05):
    movingAverage = df.rolling(window=12).mean()
    movingSTD = df.rolling(window=12).std()
    orig = plt.plot(df, color='blue', label='Original')
    mean = plt.plot(movingAverage, color='red', label='Rolling Mean')
    std = plt.plot(movingSTD, color='black', label='Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    p_value,statistic,critical_values=adfuller_test(df)
    if (p_value<p_value_check):
        print('Time series is stationary, p value:{}'.format(p_value))
    else:
        print('Time series is NOT stationary, p value:{}'.format(p_value))
    print('Statistic:{}'.format(statistic))
    for key, value in critical_values.items():
        print('Critical Values:')
        print(f'    {key}, {value}')

series = df['Adj Close']
stat_check(series)
```

Рисунок 2.2.16 – Реалізація доповненого тесту Діки-Фулера (ADF).

Обидва часові ряди показують тенденцію до зростання з виразними піками і спадами, що підтверджується як оригінальними даними (синя лінія), так і ковзним середнім (червона лінія). Середньоквадратичне відхилення (чорна лінія) залишається відносно стабільним протягом усього періоду.

Результати тесту ADF для обох часових рядів демонструють, що ряди не є стаціонарними, оскільки р-значення значно перевищує 0.05 (перший графік: р-значення = 0.4594, другий графік: р-значення = 0.9229). Це означає, що ми не можемо відхилити нульову гіпотезу про наявність одиничного кореня, що свідчить про те, що середні значення і дисперсії рядів змінюються з часом, і їх рівні залежні від попередніх значень (рис. 2.2.17 та рис. 2.2.18).

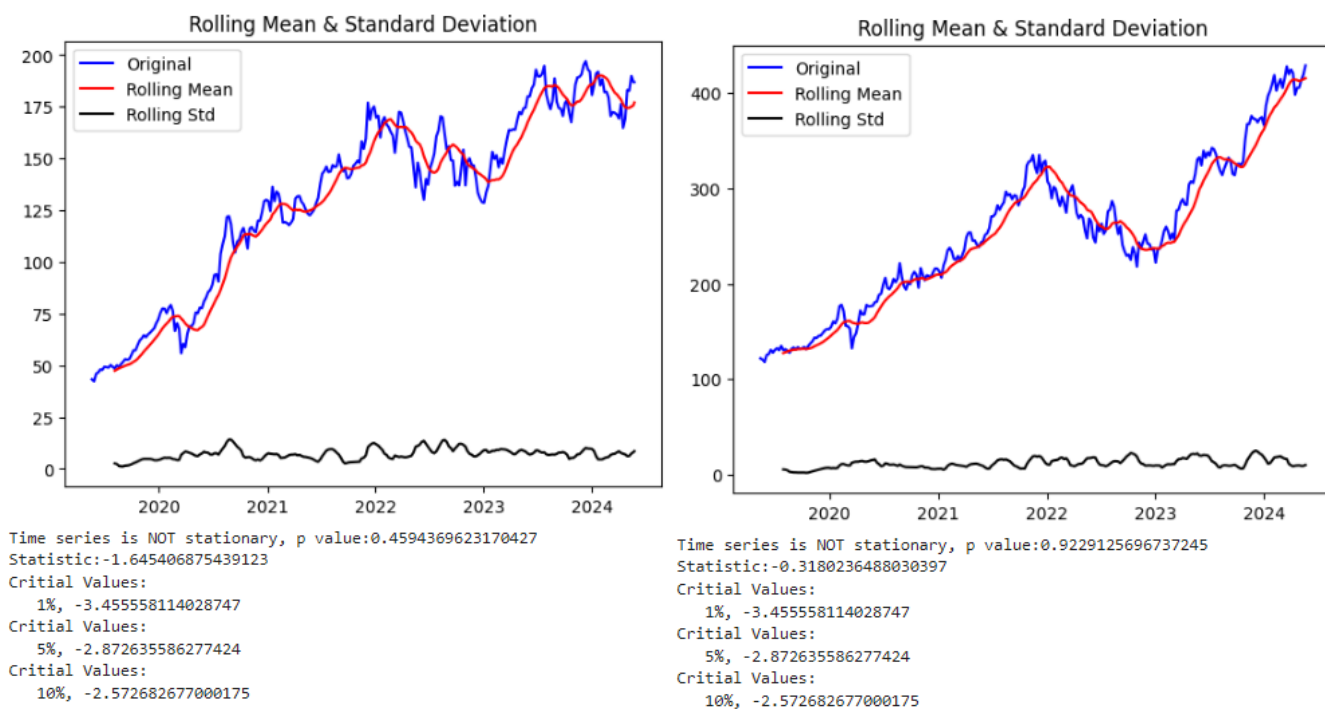


Рисунок 2.2.17 – Результат тесту Дики-Фулера (ADF),
 ліворуч AAPL, праворуч MSFT.

Автокореляція

Автокореляція – це статистична міра, яка показує, як поточне значення часового ряду пов'язане з його попередніми значеннями. Вона допомагає виявити залежності між значеннями ряду у часі, що є важливим для аналізу і прогнозування поведінки часового ряду[31].

Побудови графіка автокореляційної функції (ACF) для ряду "Adj Close" з даних DataFrame df. Функція plot_acf обчислює і відображає автокореляцію на різних лагах (в даному випадку до 5 лагів) (рис. 2.2.19).

```
series = df['Adj Close']
plot_acf(series,lags=5)
plt.show()
```

Рисунок 2.2.19 – Побудови графіка автокореляційної функції.

Висока автокореляція на всіх лагах вказує на сильний зв'язок між значеннями ряду на різних часових інтервалах. Статистична значущість: Вертикальні лінії

виходять за межі довірчого інтервалу, що свідчить про статистичну значущість автокореляцій для всіх представлених лагів(рис. 2.2.20).

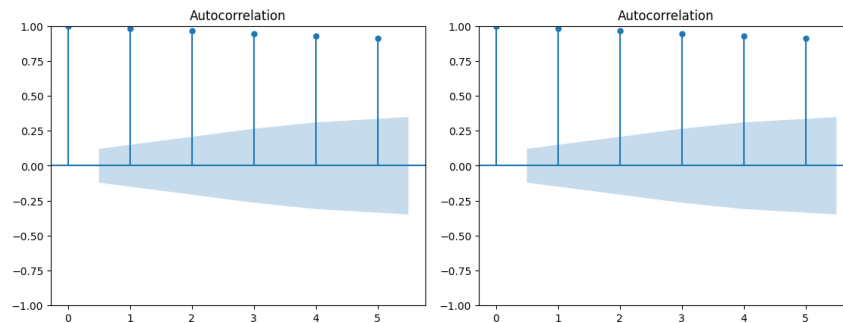


Рисунок 2.2.20 – Графіки функції (ACF), ліворуч AAPL, праворуч MSFT.

2.3 Реалізація обраного методу прогнозування цін на акції

У даному розділі викладений алгоритм що включає в себе підготовку даних, тюнінг моделі та прогнозування майбутніх значень.

Моделі на основі нейронних мереж, такі як LSTM/GRU, є досить потужними, що здатні працювати з нестационарними рядами без попередньої декомпозиції.

Для початку імпортовано додаткові бібліотеки та модулі, необхідні для підготовки даних, моделювання та роботи з нейронними мережами, включаючи моделі GRU(рис. 2.3.1):

1. Імпорт бібліотек `pandas`, `matplotlib`, `numpy`[32] та `datetime`[33] для роботи з даними, побудови графіків та обробки дат.
2. Використання `dateutil.relativedelta` для роботи з датами.
3. Імпорт `MinMaxScaler` з `sklearn.preprocessing` для масштабування даних.
4. Встановлення та імпорт бібліотеки `keras-tuner`[34] для гіперпараметричного налаштування моделей.
5. Підключення `Google Drive` для завантаження або збереження даних.

6. Імпорт модулів з keras та tensorflow[35] для створення та навчання моделей нейронних мереж, включаючи GRU (Gated Recurrent Unit) та ранню зупинку (EarlyStopping) для запобігання перенавчанню.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import datetime
from dateutil.relativedelta import relativedelta
from sklearn.preprocessing import MinMaxScaler
!pip install -q -U keras-tuner
import keras_tuner as kt
from keras_tuner import HyperModel
from google.colab import drive
drive.mount('/content/drive')
from keras.models import Sequential
from keras.layers import GRU, Dense
import tensorflow as tf
from keras.callbacks import EarlyStopping
```

Рисунок 2.3.1 – Імпортування додаткових бібліотек та модулів

Написання функції(див. додаток Г), що перетворює початковий DataFrame з цінами акцій у формат з ковзаючим вікном, що дозволяє використовувати дані для моделей машинного навчання. Основні кроки:

1. Перетворення дат: Перетворює строки з датами на об'єкти datetime.
2. Ініціалізація: Створює списки для зберігання цільових дат, наборів даних (X) та цільових значень (Y).
3. Цикл обробки даних: Виконує цикл, у якому кожен раз обирається піднабір даних за вказаний проміжок часу (n днів) і додаються значення цін у відповідні списки.
4. Перевірка розміру вікна: Перевіряє, чи розмір вікна не перевищує доступні дані на поточну дату.
5. Перехід до наступної дати: Розраховує наступну дату для обробки, додаючи місяць до поточної дати.
6. Завершення циклу: Встановлює прапорець завершення, коли досягає останньої дати.

7. Створення нового DataFrame: Перетворює зібрані дані у новий DataFrame, де кожен рядок містить дату, значення цін за попередні дні (X) та цільове значення (Y).

За допомогою виклику `df_to_windowed_df` створюється DataFrame з ковзаючим вікном для діапазону дат від "2019-06-10" до "2024-05-23" з розміром вікна, що дорівнює 3 (рис. 2.3.2). Потім результат зберігається в змінну `wd_df`. Це дозволяє перетворити початковий набір даних у формат, де кожен рядок містить дату, значення цін за три попередні дні (X) та цільове значення (Y), що відповідає ціні на наступний день після трьох попередніх (рис. 2.3.3).

```
wd_df=df_to_windowed_df(df, "2019-06-10", "2024-05-23", 3)
wd_df
```

Рисунок 2.3.2 – Виклик `df_to_windowed_df`

	Target Date	Target-3	Target-2	Target-1	Target
0	2019-06-10	43.180859	42.239899	45.878304	46.503212
1	2019-06-17	42.239899	45.878304	46.503212	47.960503
2	2019-06-24	45.878304	46.503212	47.960503	47.753014
3	2019-07-01	46.503212	47.960503	47.753014	49.275448
4	2019-07-08	47.960503	47.753014	49.275448	49.051064

Рисунок 2.3.3 – Створений необхідний формат даних

Далі виконується розділення даних на навчальний, тестовий та валідаційний набори (рис. 2.3.4):

1. Навчальний набір (`train_df`): Перші 156 рядків `wd_df`, що використовуватимуться для тренування моделі.
2. Тестовий набір (`test_df`): Останні 104 рядки `wd_df`, за винятком останніх 52 рядків, що використовуватимуться для тестування моделі.
3. Валідаційний набір (`val_df`): Останні 52 рядки `wd_df`, що використовуватимуться для валідації моделі.

Це розділення забезпечує структуровану підготовку даних для ефективного навчання та оцінки моделі.

```
#train/test split
train_df=wd_df[:156]
test_df=wd_df[-104:-52:]#1 year for test data
val_df=wd_df[-52:]#1 year for validation
```

Рисунок 2.3.4 – Розділення даних на навчальний, тестовий та валідаційний набори

Наступна функція виконує масштабування даних за допомогою `MinMaxScaler` з бібліотеки `sklearn` (рис. 2.3.5).

1. Ініціалізація масштабувальника: Створюється об'єкт `MinMaxScaler`.
2. Копіювання `DataFrame`: Створюється копія переданого `DataFrame`, щоб уникнути змін в оригінальному наборі даних.
3. Масштабування даних: Для кожної колонки (починаючи з другої) дані перетворюються за допомогою `MinMaxScaler`.
 - Якщо `train` встановлено на `True`, використовуються методи `fit_transform` для навчальних даних.
 - Якщо `train` встановлено на `False`, використовуються методи `transform` для тестових або валідаційних даних.
4. Повернення масштабованого `DataFrame`: Повертається `DataFrame` з нормалізованими значеннями.

Це дозволяє масштабувати дані для навчальних, тестових та валідаційних наборів, забезпечуючи, що модель працює з нормалізованими значеннями.

```

#data scaling, bool "train" variable indicates the way of transforming
scaler = MinMaxScaler()
columns=train_df.columns
def scaling(df,train):
    df=df.copy()
    for i in range(1,len(columns)):
        arr=np.asarray(df[columns[i]]).reshape(-1, 1)
        #for train data
        if(train==True):
            arr_norm=scaler.fit_transform(arr)
        #for test/val
        else:
            arr_norm=scaler.transform(arr)
        df[columns[i]]=arr_norm
    return df

```

Рисунок 2.3.5 – Функція виконує масштабування даних

Наступним кроком виконується масштабування даних для навчального, тестового та валідаційного наборів даних (рис. 2.3.6):

1. Навчальний набір (train_norm_df): Масштабування за допомогою функції scaling з параметром True, що означає, що використовуються методи fit_transform для навчальних даних.
2. Тестовий набір (test_norm_df): Масштабування за допомогою функції scaling з параметром False, що означає, що використовуються методи transform для тестових даних.
3. Валідаційний набір (val_norm_df): Масштабування за допомогою функції scaling з параметром False, що означає, що використовуються методи transform для валідаційних даних.

Після масштабування, значення в усіх наборах даних нормалізуються, що дозволяє моделі працювати з більш однорідними та порівнянними даними(рис. 2.3.7).

```

train_norm_df=scaling(train_df,True)
test_norm_df=scaling(test_df,False)
val_norm_df=scaling(val_df,False)
val_norm_df

```

Рисунок 2.3.6 – Масштабування даних

	Target Date	Target-3	Target-2	Target-1	Target
208	2023-06-05	0.979134	0.981193	1.023283	1.023360
209	2023-06-12	0.981193	1.023283	1.023360	1.053555
210	2023-06-19	1.023283	1.023360	1.053555	1.066975
211	2023-06-26	1.023360	1.053555	1.066975	1.122562

Рисунок 2.3.7 – Фрагмент результату масштабування

Наступна функція перетворює DataFrame з ковзаючим вікном у три окремі масиви: дати, X та Y, за таким алгоритмом (рис. 2.3.8):

1. Перетворення DataFrame в масив: перетворює DataFrame у numpy-масив за допомогою `to_numpy()`.
2. Виділення дат: збирає першу колонку масиву (дата) в окремий масив `dates`.
3. Створення матриці середніх значень: виділяє всі колонки з другої по передостанню, щоб створити середню матрицю `middle_matrix`.
4. Формування X: перетворює середню матрицю у тривимірний масив X з розмірами (кількість дат, кількість ознак, 1).
5. Формування Y: виділяє останню колонку масиву (цільове значення) в окремий масив Y.
6. Повернення значень: повертає масиви `dates`, X та Y, з приведенням X та Y до типу `np.float32`.

Окремі масиви можуть бути використані для навчання та оцінки моделей.

```
def windowed_df_to_date_X_y(windowed_dataframe):
    df_as_np = windowed_dataframe.to_numpy()
    dates = df_as_np[:, 0]
    middle_matrix = df_as_np[:, 1:-1]
    X = middle_matrix.reshape((len(dates), middle_matrix.shape[1], 1))
    Y = df_as_np[:, -1]
    return dates, X.astype(np.float32), Y.astype(np.float32)
```

Рисунок 2.3.8 – Функція перетворює DataFrame з ковзаючим вікном

Далі використовується функція `windowed_df_to_date_X_y` для розділення нормалізованих навчального, тестового та валідаційного наборів даних на три окремі масиви: дати, X та Y (рис. 2.3.9). Це забезпечує структуру даних, необхідну для навчання та оцінки моделей машинного навчання.

```
#split on dates, x and y
dates_train, X_train, y_train = windowed_df_to_date_X_y(train_norm_df)
dates_test, X_test, y_test = windowed_df_to_date_X_y(test_norm_df)
dates_val, X_val, y_val = windowed_df_to_date_X_y(val_norm_df)
```

Рисунок 2.3.9 – Функція `windowed_df_to_date_X_y` для розділення наборів даних

Візуалізуємо цільові значення (Y) для навчального, тестового та валідаційного наборів даних на одному графіку, код також виводить форми (розміри) масивів X та Y для кожного набору (рис. 2.3.10). Це дозволяє візуально оцінити розподіл цільових значень у кожному наборі даних, а також перевірити правильність формування даних (рис. 2.3.11).

```
plt.plot(dates_train, y_train)
plt.plot(dates_test, y_test)
plt.plot(dates_val, y_val)

plt.legend(['Train', 'Test', 'Val'])
print("X_train shape:",X_train.shape,"Y_train shape:",y_train.shape,
      "X_test shape:",X_test.shape,"Y_test shape:",y_test.shape,
      "X_val shape:",X_val.shape,"Y_val shape:",y_val.shape)
```

Рисунок 2.3.10 – Код для візуалізації наборів даних

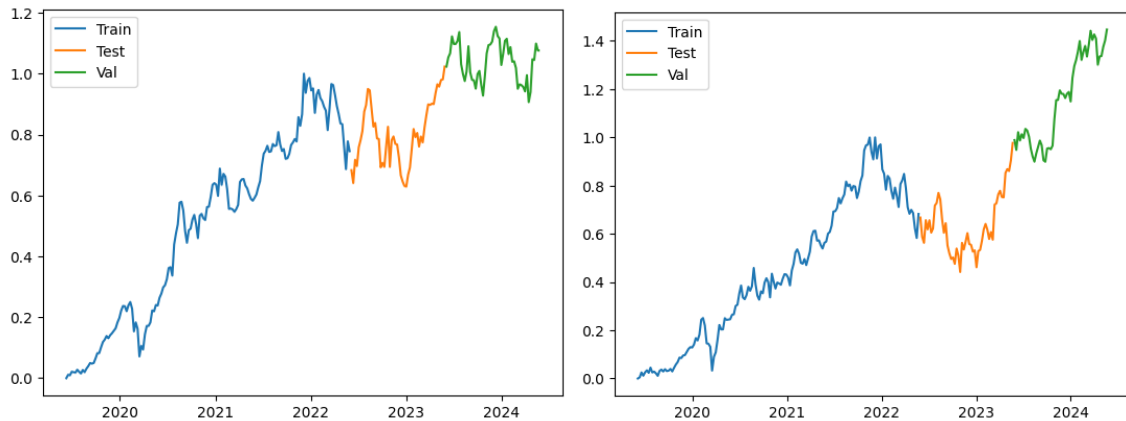


Рисунок 2.3.11 – Візуалізація цільових значень для навчального, тестового та валідаційного наборів даних, ліворуч AAPL, праворуч MSFT.

Код (див. додаток Д), створює клас моделі `MyHyperModel`, який використовує гіперпараметри для налаштування архітектури нейронної мережі за наступним алгоритмом:

1. Ініціалізація класу `MyHyperModel`:

- Метод `__init__` зберігає форму вхідних даних (`input_shape`).

2. Метод `build`:

- Створює послідовну модель `Sequential`.
- Додає перший шар `GRU` з параметрами:
 - `units`: кількість нейронів (від 16 до 256 з кроком 16).
 - `dropout`: ймовірність відсіву нейронів (від 0.0 до 0.5 з кроком 0.1).
 - `activation`: функція активації ("relu").
 - `return_sequences`: повертає послідовності, щоб передати їх наступному шару.
 - `input_shape`: форма вхідних даних.
- Додає додаткові шари `GRU` в циклі:
 - Кількість шарів визначається гіперпараметром `num_layers` (від 1 до 5).
 - Кожен шар має свої параметри `units` та `dropout`.
 - `return_sequences` встановлюється у `True` для всіх шарів, окрім останнього.

- Додає вихідний шар Dense з 1 нейроном для передбачення.
- Визначає гіперпараметр `learning_rate` для оптимізатора (значення: $1e-2$, $1e-3$, $1e-4$).
- Компілює модель з використанням оптимізатора Adam та функції втрат (гіперпараметр `loss`: "mae" або "mse").

3. Створення екземпляра `hypermodel`:

- Ініціалізується модель з вхідною формою (3, 1).

Алгоритм дозволяє створювати та налаштовувати модель з різними гіперпараметрами для оптимального навчання.

Наступним кроком використовується Hyperband для пошуку найкращих гіперпараметрів моделі `MuHyperModel` з використанням бібліотеки Keras Tuner (рис. 2.3.12).

Основні етапи:

1. Створення тюнера Hyperband:

- `objective='val_loss'`: Оптимізує втрати на валідаційному наборі.
- `max_epochs=15`: Максимальна кількість епох для кожної ітерації.
- `factor=3`: Фактор зменшення кількості ресурсів на кожному кроку.
- `directory` та `project_name`: Вказують директорію для збереження результатів та ім'я проєкту.
- `overwrite=True`: Перезаписує існуючі результати проєкту.

2. Використання `EarlyStopping` для запобігання перенавчанню:

- `monitor="val_loss"`: Моніторить втрати на валідаційному наборі.
- `patience=7`: Зупиняє навчання, якщо втрати на валідаційному наборі не покращуються протягом 7 епох.

3. Пошук гіперпараметрів:

- `tuner.search`: Виконує пошук гіперпараметрів з використанням навчальних та тестових даних.
- `epochs=100`: Максимальна кількість епох для пошуку.

- `batch_size=8`: Розмір пакету.
- `callbacks=[callback]`: Використовує раніше створений `EarlyStopping` для прискорення пошуку.

4. Отримання найкращих гіперпараметрів:

- `best_hps = tuner.get_best_hyperparameters(num_trials=1)[0]`: Отримує найкращі гіперпараметри з усіх спроб.
- `print("best model params:")`: Виводить найкращі гіперпараметри.

Це дозволяє автоматично налаштувати модель, знаходячи оптимальні гіперпараметри для найкращої продуктивності.

```
tuner = kt.Hyperband(hypermodel,
                    objective='val_loss',
                    max_epochs=15,
                    factor=3,
                    directory="/content/drive/My Drive/Model_tuning1/",
                    project_name="model_tuning",
                    overwrite=True)

#use EarlyStopping to prevent overfinng and speed up hyperparameters search
callback = EarlyStopping(
    monitor="val_loss",
    patience=7
)

tuner.search(X_train, y_train, validation_data=(X_test, y_test), epochs=100, batch_size=8, callbacks=[callback])

# Get best hyperparameters
best_hps = tuner.get_best_hyperparameters(num_trials=1)[0]
print("best model params:")
best_hps.values
```

Рисунок 2.3.12 – Використовується `Hyperband` для пошуку найкращих гіперпараметрів

Код налаштовує та тренує модель з найкращими знайденими гіперпараметрами, зберігаючи найкращі ваги та використовуючи ранню зупинку для запобігання перенавчанню (рис. 2.3.13).

Основні кроки:

1. Визначення шляху для збереження ваг:

- `weight_path='/content/drive/My Drive/checkpoint.model.h5'`: Встановлює шлях для збереження найкращих ваг моделі.

2. Створення зворотних викликів:

- `ModelCheckpoint`: Зберігає найкращі ваги моделі на основі валідаційних втрат.
 - `monitor="val_loss"`: Моніторить втрати на валідаційному наборі.
 - `mode="min"`: Зберігає модель з мінімальними втратами.
 - `save_best_only=True`: Зберігає тільки найкращі ваги.
 - `verbose=7`: Виводить повідомлення про збереження ваг.
 - `EarlyStopping`: Запобігає перенавчанню.
 - `monitor="val_loss"`: Моніторить втрати на валідаційному наборі.
 - `patience=7`: Зупиняє навчання, якщо втрати на валідаційному наборі не покращуються протягом 7 епох.
3. Побудова моделі з найкращими гіперпараметрами:
- `model = tuner.hypermodel.build(best_hps)`: Створює модель з найкращими знайденими гіперпараметрами.
4. Навчання моделі:
- `model.fit`: Тренує модель на навчальному наборі.
 - `epochs=100`: Максимальна кількість епох для навчання.
 - `validation_data=(X_test, y_test)`: Тестовий набір даних для валідації.
 - `batch_size=8`: Розмір пакету.
 - `callbacks=[callback, checkpoint]`: Використовує зворотні виклики для ранньої зупинки та збереження найкращих ваг.
5. Завантаження найкращих ваг:
- `model.load_weights("/content/drive/My Drive/checkpoint.model.h5")`: Завантажує найкращі збережені ваги у модель.

```

weight_path='/content/drive/My Drive/checkpoint.model.h5'

#save best weights to the checkpoint, if use callback for this task incorrect weights won't be stored correctly
checkpoint = tf.keras.callbacks.ModelCheckpoint(weight_path,
                                                monitor="val_loss", mode="min",
                                                save_best_only=True, verbose=7)

callback = EarlyStopping(
    monitor="val_loss",
    patience=7
)

model = tuner.hypermodel.build(best_hps)
history=model.fit(X_train, y_train, epochs=100,validation_data=(X_test, y_test), batch_size=8,callbacks=[callback,checkpoint])
#load best weights to the model
model.load_weights("/content/drive/My Drive/checkpoint.model.h5")

```

Рисунок 2.3.13 – Код для налаштування тренування

Використаємо функція `plot_history` для побудови графіків зміни втрат моделі під час навчання. Ця функція приймає об'єкт `history`, який містить інформацію про втрати на тренувальній та валідаційній вибірках для кожної епохи (рис. 2.3.14).

Результат виконання див на рисунку 3.1.1.

```

def plot_history(history):
    print(history.history.keys())
    # "Loss"
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'validation'], loc='upper left')
    plt.show()
plot_history(history)

```

Рисунок 2.3.14 – Функція `plot_history`

Додаємо функцію `calculate_mape` яка обчислює середню абсолютну відносну похибку (МАРЕ) між фактичними та передбаченими значеннями (рис. 2.3.15).

```

def calculate_mape(actual, predicted):

    if([isinstance(actual, np.ndarray) or isinstance(predicted, np.ndarray)]):
        actual, predicted = np.array(actual), np.array(predicted)
    else:
        actual, predicted = np.Int(actual), np.Int(predicted)

    return round(np.mean(np.abs((
        actual - predicted) / actual)) * 100, 2)

```

Рисунок 2.3.15 – Функцію `calculate_mape`

Далі було виконано прогнозування за допомогою LSTM моделі, розраховано MAPE, а також побудовано графік порівняння фактичних та передбачених значень (рис. 2.3.16). Ось опис кожного кроку:

1. Виконано прогнозування на валідаційних даних за допомогою моделі LSTM і перетворено результати у плоский масив.
2. Розраховано MAPE між фактичними (`y_val`) та передбаченими (`lstm_predictions`) значеннями.
3. Виведено значення MAPE.
4. Створено функцію `plot_predictions`, яка буде графік фактичних та передбачених значень (рис. 2.3.17).

```
lstm_predictions = model.predict(X_val).flatten()
lstm_score = 100-calculate_mape(y_val, lstm_predictions)
print("MAPE of LSTM model = ",lstm_score)
def plot_predictions(test, predicted, title):
    plt.figure(figsize=(16,4))
    plt.plot(test, color='blue',label='Actual')
    plt.plot(predicted, alpha=0.7, color='orange',label='Predicted')
    plt.title(title)
    plt.xlabel('Time')
    plt.ylabel('Normalized scale')
    plt.legend()
    plt.show()

plot_predictions(y_val, lstm_predictions, "Predictions made by Bidirectional GRU model")
```

Рисунок 2.3.16 – Виконання прогнозування

```
prediction_df=pd.DataFrame({"Prediction":np.asarray(scaler.inverse_transform(lstm_predictions.reshape(-1, 1))).reshape(52),
                           "True value":np.array(df["Adj Close"][-52:]),index=dates_val)
prediction_df["Prediction"]=[round(value) for value in prediction_df["Prediction"]]
prediction_df
```

Рисунок 2.3.17 – Функція `plot_predictions`

РОЗДІЛ 3 ОПИС РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ ТА ЇХ ВЕРИФІКАЦІЯ

3.1 Опис результатів моделювання та прогнозування цін акцій

Провівши послідовне виконання вищезгаданого коду отримуємо наступні результати:

Аналіз отриманих графіків зміни втрат показує(рис. 3.3.1), що втрата для тренувального набору (синя лінія) і валідаційного набору (помаранчева лінія) швидко зменшується протягом перших кількох епох і стабілізується після приблизно п'ятої епохи. Це вказує на ефективне навчання моделі, де після початкової швидкої адаптації модель досягає стабільного стану з мінімальними втратами. Відсутність значного розриву між тренувальними та валідаційними втратами свідчить про те, що модель не перенавчається і добре узагальнює дані((рис. 3.3.1)ліворуч AAPL).

Другий графік ((рис. 3.1.1)праворуч MSFT) також демонструє швидке зменшення втрат на початкових етапах тренування, причому як тренувальна, так і валідаційна втрати досягають дуже низьких значень вже після третьої епохи. В подальшому обидві криві залишаються низькими та стабільними, що вказує на хорошу якість моделі та її здатність узагальнювати нові дані без ознак перенавчання та недонавчання.

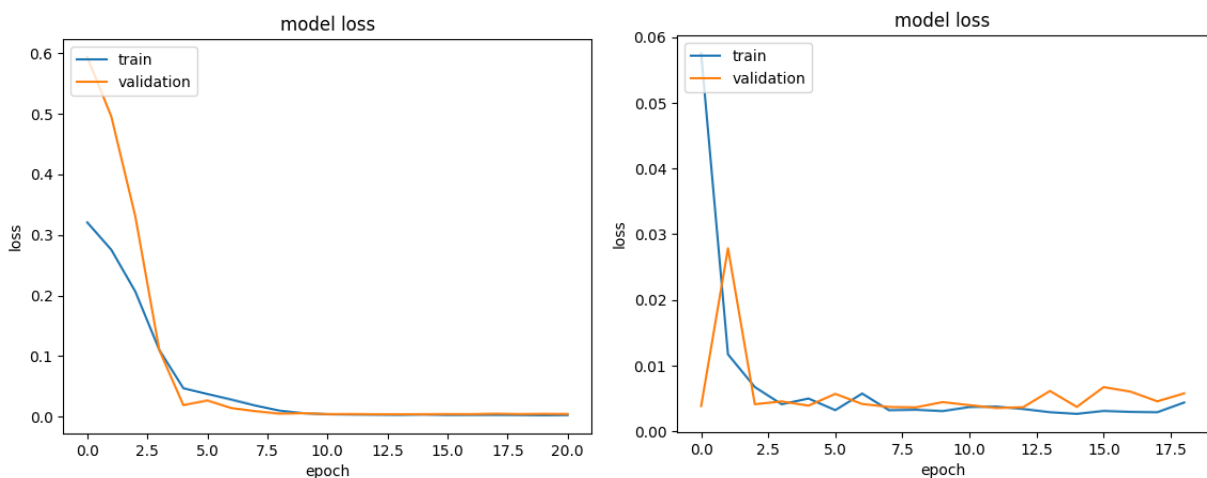


Рисунок 3.1.1 – Графіки зміни втрат моделі під час навчання, ліворуч AAPL, праворуч MSFT.

Розглянемо результат прогнозування за допомогою LSTM моделі, а також побудований графік порівняння фактичних та передбачених значень.

Модель прогнозування акції AAPL добре відображає загальні тенденції і коливання фактичних даних, хоча є певні відхилення в конкретних точках. Це свідчить про те, що модель успішно вловлює основні патерни в даних, але може мати проблеми з точним передбаченням значень в окремих випадках. В цілому, передбачені значення слідують за фактичними змінами, що свідчить про здатність моделі до адекватного прогнозування динаміки часового ряду (рис. 3.1.2).

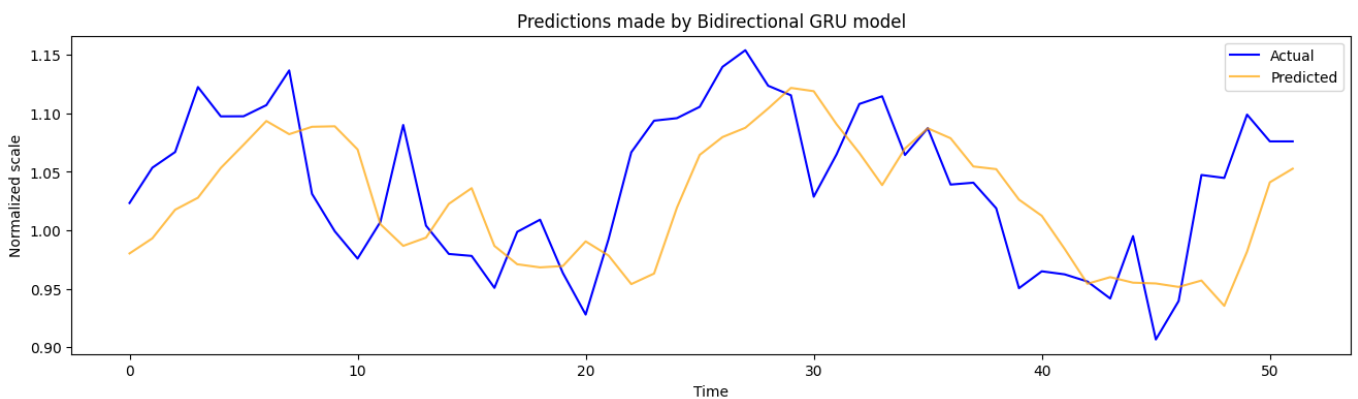


Рисунок 3.1.2 – Результат прогнозування AAPL.

Прогнозування акції MSFT в загальному випадку вловлює тенденції зростання та зниження фактичних даних. Проте, є помітні розбіжності між фактичними і передбаченими значеннями, особливо в точках з різкими змінами. Фактичні значення мають більшу волатильність і більш виражені піки порівняно з передбаченими значеннями, які показують більш згладжену динаміку.

Це свідчить про те, що модель в загальному добре справляється з прогнозуванням довгострокових тенденцій, але може не вловлювати короткострокові флуктуації з достатньою точністю. Для покращення точності прогнозів може знадобитися додаткове налаштування моделі, збільшення кількості тренувальних даних або використання інших методів.

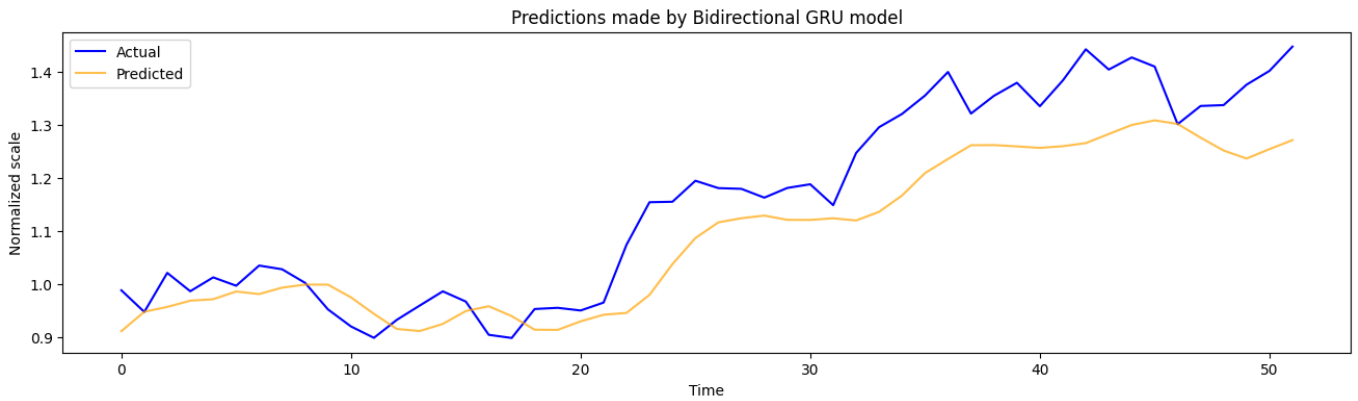


Рисунок 3.1.3 – Результат прогнозування MSFT.

3.2 Верифікація прогнозів

Верифікація якості прогнозів є ключовим аспектом у процесі оцінювання та вдосконалення моделей прогнозування. Вона дозволяє оцінити, наскільки точними є прогнози, зроблені моделлю, порівнюючи їх з фактичними значеннями. Для цього використовуються різні метрики, серед яких найбільш поширеними є середньоквадратична похибка (MSE), середня абсолютна похибка (MAE) та середня абсолютна відносна похибка (MAPE).

MSE визначається як середнє значення квадратів різниць між прогнозованими та фактичними значеннями. Але надає високі значення помилкам, що значно відхиляються від фактичних значень, оскільки помилка підноситься до квадрату. Це робить MSE чутливим до великих помилок, що є корисним у випадках, коли великі помилки є небажаними.

MAE є більш інтуїтивно зрозумілою метрикою, оскільки вона відображає середнє значення помилки в тих самих одиницях вимірювання, що й оригінальні дані. MAE менш чутлива до великих помилок порівняно з MSE, що робить її корисною в ситуаціях, коли важливо оцінити загальну точність прогнозу без надмірної уваги до великих відхилень.

МАРЕ визначається як середнє значення абсолютних відносних похибок між прогнозованими та фактичними значеннями, виражене у відсотках.

Розрахуємо показники для отриманих прогнозів(рис. 3.2.1 та рис. 3.2.2).

```
print("MSE:",float(tf.keras.metrics.mean_squared_error(prediction_df["True value"],prediction_df["Prediction"])))
print("MAE:",float(tf.keras.metrics.mean_absolute_error(prediction_df["True value"],prediction_df["Prediction"])))
print("MAPE (Accuracy):",100- calculate_mape(np.array(prediction_df["True value"]),np.array(prediction_df["Prediction"])))
gru_mape=calculate_mape(np.array(prediction_df["True value"]),np.array(prediction_df["Prediction"]))
```

```
MSE: 57.0
MAE: 6.0
MAPE (Accuracy): 96.53
```

Рисунок 3.2.1 – Результат показників точності для AAPL.

```
print("MSE:",float(tf.keras.metrics.mean_squared_error(prediction_df["True value"],prediction_df["Prediction"])))
print("MAE:",float(tf.keras.metrics.mean_absolute_error(prediction_df["True value"],prediction_df["Prediction"])))
print("MAPE (Accuracy):",100- calculate_mape(np.array(prediction_df["True value"]),np.array(prediction_df["Prediction"])))
gru_mape=calculate_mape(np.array(prediction_df["True value"]),np.array(prediction_df["Prediction"]))
```

```
MSE: 359.0
MAE: 15.0
MAPE (Accuracy): 95.8
```

Рисунок 3.2.2 – Результат показників точності для MSFT.

Оскільки розраховувалось МАРЕ(Accuracy), то МАРЕ визначимо як 100-МАРЕ(Accuracy), щоб дізнатися відсоток похибки а не точності.

Виходячи з показників можна зробити висновок, що модель прогнозування цін акцій демонструє хорошу точність. Показник похибки МАРЕ 3.47% для акцій AAPL та 4.2% для MSFT вказує на високу загальну точність моделей. Проте, значення MSE та MAE для MSFT значно вищі порівняно з AAPL, що може свідчити про необхідність потенційних вдосконалень у моделі прогнозування для акцій MSFT. Зокрема, можливе уточнення вибірки даних або корекція методів машинного навчання для зменшення помилок прогнозування. Водночас, низькі значення MAE та MSE для акцій AAPL вказують на відносно високу ефективність.

ВИСНОВОКИ

Фондові ринки та цінні папери відіграють ключову роль у глобальній економіці, забезпечуючи капітал для підприємств і державних установ та надаючи інвесторам можливості для збільшення їхніх активів. В умовах сучасної світової економіки, де економічні процеси дедалі більше глобалізуються та інтегруються, фондові ринки стають індикаторами економічного здоров'я країн та регіонів. Вони відображають настрої інвесторів, очікування щодо економічного зростання та фінансової стабільності. Крім того, фондові ринки сприяють ефективному розподілу ресурсів, підтримуючи інновації та розвиток технологій.

Систематичне дослідження фондових ринків і цінних паперів є надзвичайно важливим через їхню складність та динамічність. Аналіз поведінки фондових ринків, оцінка ризиків та прогнозування цінових змін потребують використання передових математичних та статистичних методів. Враховуючи непередбачуваність і волатильність ринків, регулярний моніторинг та дослідження стають необхідними для забезпечення стабільності та уникнення фінансових криз. Крім того, сучасні виклики, такі як вплив пандемій, політичних нестабільностей та екологічних катастроф, роблять дослідження у сфері фінансових ринків ще більш актуальним та необхідним для адаптації до нових умов і збереження економічної безпеки.

У процесі дослідження було здійснено глибокий аналіз факторів, що впливають на цінову динаміку акцій, а також розроблено методи прогнозування. Зокрема, розглянуто роль фундаментального та технічного аналізу. Фундаментальний аналіз дозволяє оцінити вартість акцій на основі фінансового стану компанії, макроекономічних показників і галузевих тенденцій. Натомість, технічний аналіз базується на вивченні історичних даних про ціни та обсяги торгівлі для виявлення трендів і закономірностей. Виявлено, що поєднання цих двох підходів дає змогу більш точно оцінювати ринкові умови та прогнозувати цінові рухи.

Важливою частиною роботи стало вивчення впливу новин та суспільних подій на ринки. Аналіз новинного фону показав, що значущі економічні події, політичні рішення та інші макроекономічні фактори можуть мати істотний вплив на настрої інвесторів та відповідно на цінові динаміки акцій. Дослідження також охоплювало аналіз настроїв інвесторів, що дозволяє зрозуміти, як позитивні або негативні новини відображаються на ринку. Використання сучасних інструментів для аналізу новин і соціальних медіа може бути ефективним у прогнозуванні короткострокових ринкових змін.

Останній розділ дослідження був присвячений використанню машинного навчання для прогнозування цін акцій. В роботі розглянуто різні типи алгоритмів машинного навчання, включаючи алгоритми прогнозування часових рядів. Реалізація обраного методу прогнозування на основі машинного навчання продемонструвала хорошу точність у передбаченні цінових змін. Однак дослідження також виявило ризики та обмеження таких моделей, зокрема їхню чутливість до якості даних та можливість перенавчити, може виникнути упередженість чи недонавчання. Верифікація прогнозів показала, що використання передових методів аналізу та прогнозування дозволяє знизити невизначеність та підвищити ефективність інвестиційних стратегій.

Отримані результати можуть бути покращені та використані для удосконалення інвестиційних стратегій як окремих інвесторів, так і фінансових установ. Використання моделей машинного навчання дозволяє автоматизувати процес прогнозування, що значно підвищує оперативність та точність інвестиційних рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fundamental analysis. Corporate Finance Institute. URL: <https://corporatefinanceinstitute.com/resources/valuation/fundamental-analysis/> (date of access: 03.04.2024).

2. Макроекономічні показники фундаментального аналізу. TradingInfo. URL: <https://tradinginfo.club/макроекономічні-показники-фундамент/> (дата звернення: 04.04.2024).

3. Фундаментальний аналіз. CRYPTOLOGY. URL: <https://cryptology.school/blog/shho-take-fundamentalnii-analiz-i-dlia-cogo-vin-potriben> (дата звернення: 04.04.2024).

4. Сучасні аспекти аналізу інвестиційної привабливості. Ефективна економіка. URL: <http://www.economy.nayka.com.ua/?op=1&z=2165> (дата звернення: 07.04.2024).

5. Chen J. Technical analysis of stocks and trends definition. Investopedia. URL: <https://www.investopedia.com/terms/t/technical-analysis-of-stocks-and-trends.asp> (date of access: 08.04.2024).

6. Technical indicator. Corporate Finance Institute. URL: <https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/technical-indicator/> (date of access: 10.04.2024).

7. Team T. I. Patterns vs. trends: what's the difference?. Investopedia. URL: <https://www.investopedia.com/ask/answers/010715/what-are-differences-between-patterns-and-trends.asp> (date of access: 11.04.2024).

8. Dow theory: definition, analysis, trends, & principles. smallcase. URL: <https://www.smallcase.com/learn/what-is-dow-theory/> (date of access: 13.04.2024).

9. Elliott wave theory: introduction to the wave principle. Elliott Wave International. URL: <https://www.elliottwave.com/free/introduction-to-the-wave-principle/> (date of access: 14.04.2024).

10. Disadvantages of trading technical analysis | trading biased opinion. Enrich Money. URL: <https://enrichmoney.in/knowledge-center-chapter/what-are-the-disadvantages-of-technical-analysis> (date of access: 16.04.2024).

11. Заробіток на новинах на фондовому ринку. Freedom Broker. URL: <https://ffin.ua/blog/articles/investopediia/post/yak-zarobiti-na-novinaх-na-fondovomu-rinku> (дата звернення: 18.04.2024).

12. How might political events influence markets. Pacific Investment Management Company LLC. URL: <https://www.pimco.com.au/en-au/marketintelligence/investing-in-todays-markets/how-might-political-events-influence-markets> (date of access: 19.04.2024).

13. Акції vs пандемія: як фондовий ринок реагує на кризи. Mind.ua. URL: <https://mind.ua/publications/20213210-akciyi-vs-pandemiya-yak-fondovij-rinok-reague-na-krizi> (дата звернення: 20.04.2024).

14. Academy B. Ринкові настрої криптовалют. Binance Academy. URL: <https://academy.binance.com/uk/articles/what-is-crypto-market-sentiment> (дата звернення: 21.04.2024).

15. Sentiment analysis on social media: leveraging insights for data-driven decision making - aim technologies. Aim Technologies. URL: <https://www.aimtechnologies.co/sentiment-analysis-on-social-media-leveraging-insights-for-data-driven-decision-making/> (date of access: 21.04.2024).

16. Fear and greed index explained. Capital. URL: <https://capital.com/fear-and-greed-index> (date of access: 23.04.2024).

17. Call та Put на фондовому ринку. Freedom Finance. URL: <https://ffin.ua/blog/articles/investopediia/post/shho-take-call-ta-put-na-fondovomu-rinku> (дата звернення: 23.04.2024).

18. Автоматизований контент-аналіз: використання багатства текстових даних. mindthegraph. URL: <https://mindthegraph.com/blog/uk/automated-content-analysis/> (дата звернення: 25.04.2024).

19. МАШИННЕ НАВЧАННЯ: МЕТОДИ ТА МОДЕЛІ : підручник. Харків, 2020. 304 с. URL: <https://comsys.kpi.ua/upload/Rainforcement%20Learning%20.pdf> (дата звернення: 25.04.2024).

20. Machine learning, ML. IT-Enterprise. URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning> (дата звернення: 27.04.2024).
21. Переваги, які машинне навчання може дати бізнесу. Management. URL: <https://www.management.com.ua/notes/machine-learning-for-business.html> (дата звернення: 28.04.2024).
22. Time series forecasting algorithms. iunera. URL: <https://www.iunera.com/kraken/fabric/top-5-common-time-series-forecasting-algorithms/> (date of access: 29.04.2024).
23. Тренди глибокого навчання і нейронних мереж на 2024. merehead. URL: <https://merehead.com/ua/blog/deep-learning-trends-in-2024/> (дата звернення: 01.05.2024).
24. Довідник по Machine Learning – Gradient Descent For Machine Learning. База знань ІТ технологій. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/gradient-descent-for-machine-learning> (дата звернення: 02.05.2024).
25. Переваги та недоліки машинного навчання. Carlos Barraza. URL: <https://barrazacarlos.com/uk/переваги-та-недоліки-машинного-навча/> (дата звернення: 02.05.2024).
26. Yahoo Finance - Stock Market Live, Quotes, Business & Finance News. Stock Market Live, Quotes, Business & Finance News. URL: <https://finance.yahoo.com> (date of access: 07.05.2024).
27. pandas documentation – pandas 2.2.2 documentation. pandas - Python Data Analysis Library. URL: <https://pandas.pydata.org/docs/> (date of access: 07.05.2024).
28. Matplotlib documentation – Matplotlib 3.9.0 documentation. Matplotlib – Visualization with Python. URL: <https://matplotlib.org/stable/index.html> (date of access: 09.05.2024).
29. Seaborn: statistical data visualization – seaborn 0.13.2 documentation. URL: <https://seaborn.pydata.org> (date of access: 09.05.2024).
30. statsmodels 0.14.1. URL: <https://www.statsmodels.org/stable/index.html> (date of access: 11.05.2024).

31. Autocorrelation. statisticssolutions. URL: <https://www.statisticssolutions.com/dissertation-resources/autocorrelation/> (date of access: 11.05.2024).
32. NumPy Documentation. NumPy. URL: <https://numpy.org/doc/> (date of access: 12.05.2024).
33. Basic date and time types. Python documentation. URL: <https://docs.python.org/3/library/datetime.html> (date of access: 14.05.2024).
34. Keras documentation: about keras 3. Keras: Deep Learning for humans. URL: <https://keras.io/about/> (date of access: 15.05.2024).
35. Guide | TensorFlow Core. TensorFlow. URL: <https://www.tensorflow.org/guide> (date of access: 18.05.2024).

ДОДАТКИ

Додаток А

SUMMARY

Peichev I. I. Using Machine Learning to Predict Changes in Share Prices on the Stock Market. Bachelor's Qualification Work. Specialty – 051 Economics (Economic Cybernetics and business analytics) / Supervisor H. M. Yarovenko. Sumy State University, Sumy, 2024. The work explores the main aspects of stocks in the stock markets, their nature and types, forecasting methods, statistical data analysis, and current trends in the development of forecasting methods. Stock price changes were forecasted using machine learning models. The main objective of the study was to utilize and evaluate machine learning models for forecasting stock prices in the stock market.

Key words: ML, machine learning, stock, forecasting.

АНОТАЦІЯ

Пейчев І. І. Використання машинного навчання для прогнозування змін курсів акцій на фондовому ринку. Кваліфікаційна робота бакалавра. Спец. 051 – економіка(економічна кібернетика та бізнес аналітика) / наук. Кер. Г. М. Яровенко. Сумський державний університет, Суми, 2024 р.

У роботі було досліджено основні аспекти акцій на фондових ринках, їх сутність та різновиди, методи прогнозування, були проаналізовані статистичні дані та актуальні напрямки розвитку методів прогнозування. Було зпрогнозовано зміни курсу акцій за допомогою моделей машинного навчання. Основна мета дослідження полягала в використанні та оцінці моделей машинного навчання для прогнозування курсів цін акцій на фондовому ринку.

Ключові слова: ML, машинне навчання, акція, прогнозування.

Додаток Г

```

def df_to_windowed_df(dataframe, first_date_str, last_date_str, n):
    first_date = pd.to_datetime(first_date_str)
    last_date = pd.to_datetime(last_date_str)

    target_date = first_date

    dates = []
    X, Y = [], []

    last_time = False
    while True:
        df_subset = dataframe.loc[:target_date].tail(n+1)

        if len(df_subset) != n+1:
            print(f'Error: Window of size {n} is too large for date {target_date}')
            return

        values = df_subset['Adj Close'].to_numpy()
        x, y = values[:-1], values[-1]

        dates.append(target_date)
        X.append(x)
        Y.append(y)

        next_week = dataframe.loc[target_date:target_date + relativedelta(months=+1)]
        next_datetime_str = str(next_week.head(2).tail(1).index.values[0])
        next_date_str = next_datetime_str.split('T')[0]
        year_month_day = next_date_str.split('-')
        year, month, day = year_month_day
        next_date = datetime.datetime(day=int(day), month=int(month), year=int(year))
        #year_month=str(year)+"-"+str(month)
        #next_date = pd.Period(year_month, 'M')

        if last_time:
            break

        target_date = next_date

        if target_date == last_date:
            last_time = True

    ret_df = pd.DataFrame({})
    ret_df['Target Date'] = dates

    X = np.array(X)
    for i in range(0, n):
        X[:, i]
        ret_df[f'Target-{n-i}'] = X[:, i]

    ret_df['Target'] = Y

    return ret_df

```

Рисунок Г.1 – функція для перетворення DataFrame у формат з ковзаючим вікном

Додаток Д

```

#model class
class MyHyperModel(HyperModel):
    def __init__(self, input_shape):
        self.input_shape = input_shape

    def build(self, hp):
        model = Sequential()

        # Add a first LSTM layer
        model.add(GRU(
            units=hp.Int("units_first_layer", min_value=16, max_value=256, step=16),
            dropout=hp.Float("dropout_first_layer", min_value=0.0, max_value=0.5, step=0.1),
            activation="relu",
            return_sequences=True,
            input_shape=self.input_shape))

        # Add additional LSTM layers
        for i in range(1, hp.Int("num_layers", 1, 5)):
            model.add(GRU(
                units=hp.Int("units_" + str(i), min_value=16, max_value=256, step=16),
                dropout=hp.Float("dropout_" + str(i), min_value=0.0, max_value=0.5, step=0.1),
                activation="relu",
                return_sequences=(i < hp.Int("num_layers", 1, 5) - 1)))

        #model.add(Dense(units=hp.Int("units_dense_1_layer", min_value=0, max_value=128, step=16),activation="relu"))
        model.add(Dense(units=1))
        hp_learning_rate = hp.Choice('learning_rate', values=[1e-2, 1e-3, 1e-4])
        # Compile model
        model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=hp_learning_rate),
            loss=hp.Choice("loss",values=[ "mae","mse"]))

        return model

hypermodel = MyHyperModel(input_shape=(3,1))

```

Рисунок Д.1 – клас моделі MyHyperModel