КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра технологій управління

Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

"Аналіз і прогнозування котувань на фондовій біржі методами машинного навчання "

Студента 2-го курсу групи ІАВ-21

Аблаєв Руслан Баходировича
(прізвище, ім'я, по батькові)

(підпис студента)

Науковий керівник:

Кандидат технічних наук, асистент
(науковий ступінь, вчене звання)

Хлевний Андрій Олександрович
(прізвище, ім'я, по батькові)

(дата) (підпис)

Попередній захист:

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри технологій управління

(підпис) (прізвище, ініціали) (дата)

Київ – 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 - Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

_____
«____» _____20___року

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент Аблаєв Руслан Баходирович
Група___ІАВ-21_____

1. Тема кваліфікаційної роботи
Аналіз і прогнозування котувань на фондовій біржі методами машинного навчання

Затверджена наказом по від «___»_____20__ р. № ___.
2. Строк подання студентом готової роботи – "___"_____ 20___ р.

3. Цільова установка та вихідні дані до роботи: дослідження теоретичних основ ринку акцій ті біржі та застосування інформаційної аналітики у даній сфері; аналіз методів та моделей аналізу та моделювання процесів, включно з програмними засобами реалізації; побудова моделей прогнозування та її оцінка; проведення аналізу і для поадльшої імплементації отриманої моделі на підприємствах.

4. Зміст роботи: Аналіз задач прогнозування фінансових котувань, методи машинного навчання для прикладних задач прогнозування фінансових котирувань, постановка задачі аналізу і прогнозування котувань акцій на фондовому ринку методами машинного навчання, аналіз актуальності застосування методів машинного навчання для прогнозування котувань акцій на фондовому ринку, математична модель штучних нейронних мереж, математична модель методу опорних векторів, формування бази даних котувань акцій різних компаній, вибір засобів для реалізації моделі, підготовка середовища та серверної частини до програмування штучної нейронної мережі та методу опорних векторів, моделі машинного навчання для прогнозування котувань акцій, створення системи розсилки сповіщень користувачам, регулярний збір даних від yahoo finance, застосування розробленої моделі на комерційних підприємствах, потенціал розвитку досліджень і реалізацій проекту

5. Перелік графічного матеріалу (слайдів): Схема одношарової нейронної мережі, Штучний нейрон, Штучний нейрон з активаційною функцією, Сигмоїдальна логістична функція, Функція гіперболічного тангенса, Двошарова нейронна мережа, графічне представлення алгоритму SVM,

графічне представлення “викиду” і перебудовування межі, Графік котувань акцій всіх компаній, Графік котувань акцій компаній ORCL, MSFT, ADBE, Результати дослідження для компанії EPAM, Детальний розгляд прогнозу для компанії EPAM, Результати дослідження для компанії EPAM методом опорних векторів, Приклад щоденної розсилки для підтвердження якості даних.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1	Вибір теми дипломної роботи	3	01.10.2022	01.10.2022
2	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	27.10.2022	27.10.2022
3	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	08.01.2023	08.01.2023
4	Складання розгорнутого плану кваліфікаційної роботи	5	18.01.2023	18.01.2023
5	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	5	20.01.2023	20.01.2023
6	Підготовка розділу 1 “Аналіз засад використання моделей і методів машинного навчання в аналізі і прогнозуванні котувань на фондовій біржі”	10	13.02.2023	13.02.2023
7	Підготовка розділу 2 “Алгоритми машинного навчання та способи їх навчання для прогнозування котувань на фондовій біржі”	15	06.03.2023	06.03.2023
8	Підготовка розділу 3 “Побудова моделі аналізу і прогнозування котувань на фондовій біржі методами машинного навчання”	20	03.04.2023	03.04.2023
9	Підготовка розділу 4 “Застосування методів машинного навчання для аналізу та прогнозування котирувань фондового ринку”	13	17.04.2023	17.04.2023
10	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	12	01.05.2023	01.05.2023
11	Передача кваліфікаційної роботи науковому керівникові	2	02.05.2023	02.05.2023
12	Передача кваліфікаційної роботи рецензенту для рецензування	1	10.05.2023	10.05.2023
13	Попередній захист кваліфікаційної роботи	2	17.05.2023	17.05.2023

Дата видачі завдання «____» _____ 20__ р.

Керівник роботи _к.т.н., асистент Хлевний Андрій Олександрович_____
(посада, прізвище, ім’я, по батькові)

_____
(підпис)

Завдання прийняв до виконання студент групи _IAB-21____
___Аблаев Руслан Баходирович_____
(прізвище, ім’я, по батькові)

_____
(підпис)

ЗМІСТ

АНОТАЦІЯ

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра технологій управління
Спеціальність 122 – Комп’ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

Дипломна робота магістранта Аблаєва Руслана Баходировича

Тема роботи: «Аналіз і прогнозування котувань на фондовій біржі методами машинного навчання»

Мета роботи: Дослідження та аналіз можливостей застосування методів машинного навчання для аналізу та прогнозування котирувань на фондовому ринку.

Об’єкт дослідження: Фондовий ринок, котирування акцій

Предмет дослідження: Методи машинного навчання і аналіз та прогнозування котирувань на фондовому ринку.

Наукова новизна роботи "Аналіз і прогнозування котирувань на фондовій біржі методами машинного навчання" полягає в тому, що вона впроваджує інноваційні технології та методи машинного навчання для аналізу та прогнозування фондових котирувань. Використання штучних нейронних мереж (АНН) та методів опорних векторів (SVM) у фінансовому аналізі є новаторським підходом, який може покращити точність та ефективність прогнозування на фондовій біржі. Додатково, робота розробляє модель прогнозування та проводить експериментальне дослідження на реальних даних фондового ринку для оцінки ефективності розробленої моделі та порівняння з традиційними методами.

Дипломна робота складається зі вступу, основної частини, що включає 4 розділи, висновку, списку використаних джерел та додатків. Всього налічує 103 сторінок, перелік з 30 джерел на 4 сторінках та 3 додатків на 14 сторінках.

Ключові слова: машинне навчання, метод опорних векторів, штучна нейронна мережа, інформаційна аналітика даних, біржа, моделювання, котирування, прогнозування цін.

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

АНН (ANN) – Artificial Neural Network – штучна нейронна мережа;

МОВ (SVM) – Support Vector Machine – машина опорних векторів.

CNN – Convolutional Neural Network – конволюционная нейронна мережа;

DNN – Deep Neural Network – глибока нейронна мережа;

DTW – Dynamic Time Warping – динамічне викривлення часу;

LSTM – Long Short Time Memory – довготривала короткочасна пам'ять;

MLP – Many Layer Perceptron – багатошаровий персептрон;

ReLU – Rectified Linear Units – випрямлена лінійна одиниця;

MSE – Mean Square Error – середньоквадратична помилка;

RMSE – Root Mean Square Error – корінь з середньоквадратичної помилка;

ADBE – акції компанції Adobe Systems Incorporated

CSCO– акції компанції Cisco Systems, Inc.

GLOB – акції компанції Globant S.A.

GOOG – акції компанції Alphabet Inc.

MSFT – акції компанції Microsoft Corporation

ORCL – акції компанції Oracle Corporation

ВСТУП

Фондовий ринок відіграє ключову роль у світовій економіці, пропонуючи компаніям та інвесторам платформу для торгівлі цінними паперами та визначення вартості активів. Здатність аналізувати та прогнозувати котирування на фондовому ринку є надзвичайно важливою для прийняття обґрунтованих інвестиційних рішень, пом’якшення ризиків та максимізації фінансових прибутків. З появою машинного навчання та його потенціалу витягувати закономірності та ідеї зі складних даних застосування методів машинного навчання для аналізу та прогнозування котирувань на фондовому ринку стало активною областю досліджень.

Ця магістерська робота спрямована на дослідження аналізу та прогнозування котирувань на фондовому ринку за допомогою методів машинного навчання. Використовуючи потужність алгоритмів машинного навчання, ми прагнемо виявити приховані закономірності, взаємозв’язки та тенденції у великій кількості доступних фінансових даних. За допомогою цього аналізу ми прагнемо надати цінну інформацію та можливості прогнозування, які можуть допомогти інвесторам, фінансовим установам і підприємствам у прийнятті стратегічних рішень і оптимізації їхніх інвестиційних портфелів.

Використання методів машинного навчання дає ряд переваг в аналізі та прогнозуванні котирувань фондового ринку. Традиційні методи, такі як статистичні моделі або фундаментальний аналіз, часто спираються на спрощені припущення, які можуть не охопити складну динаміку ринку. З іншого боку, методи машинного навчання чудово справляються з обробкою великих обсягів даних, виявленням нелінійних зв’язків і адаптацією до мінливих умов ринку. Це дозволяє нам розробляти моделі, які потенційно можуть перевершити традиційні підходи з точки зору точності та надійності.

Для досягнення наших дослідницьких цілей ми досліджуватимемо такі алгоритми машинного навчання, включаючи, але не обмежуючись ними, штучні нейронні мережі (ANN), опорні векторні машини (SVM) і ансамблеві методи. Ці алгоритми будуть навчені на історичних ринкових даних, що включають різноманітні відповідні характеристики, такі як ціни на акції, обсяги торгів, технічні індикатори та макроекономічні змінні. Шляхом ретельної оцінки та порівняння ми прагнемо визначити найефективніші моделі для аналізу та прогнозування котирувань фондового ринку.

Метою даного дослідження є дослідження та аналіз можливостей застосування методів машинного навчання для аналізу та прогнозування котирувань на фондовому ринку. Для досягнення цієї мети будуть вирішуватись наступні завдання:

1. Зібрати та підготувати відповідні дані про ринкові котирування, включаючи історичні дані про ціни акцій, обсяги торгів, технічні показники та макроекономічні змінних.
2. Вивчити і оцінити різні методи машинного навчання, зокрема штучні нейронні мережі (АНН) та методи опорних векторів (SVM), для їх використання в аналізі та прогнозуванні котирувань.
3. Розробити та реалізувати модель, яка використовує вибрані методи машинного навчання для аналізу та прогнозування котирувань на біржі.
4. Провести експериментальне дослідження та оцінку ефективності розробленої моделі на реальних даних фондового ринку.
5. Проаналізувати та проінтерпретувати отримані результати, з'ясувати прогнозну точність та придатність розробленої моделі в порівнянні з традиційними підходами.

Об’єктом дослідження в цій темі можна вважати фондовий ринок і пов’язані з ним дані, включаючи історичні дані про ціни, обсяги торгів, основні показники компанії, настрої новин та інші відповідні змінні. Аналіз цього об’єкта спрямований

на виявлення закономірностей, тенденцій і зв’язків у даних, які можуть надати розуміння поведінки цін на акції. Предметом дослідження є аналіз та прогнозування котирувань на фондовому ринку.

В ході дослідження було отримано модель, яка використовує методи машинного навчання для аналізу та прогнозування котирувань на фондовому ринку. Проведені експерименти та аналіз показали, що розроблена модель демонструє високу прогнозну точність та перевершує традиційні підходи до аналізу ринку. Результати дослідження у подальшому будуть представлені на конференціях та відзначені в наукових публікаціях.

РОЗДІЛ 1. АНАЛІЗ ЗАСАД ВИКОРИСТАННЯ МОДЕЛЕЙ І МЕТОДІВ МАШИННОГО НАВЧАННЯ В АНАЛІЗІ І ПРОГНОЗУВАННІ КОТУВАНЬ НА ФОНДОВІЙ БІРЖІ

1.1. Аналіз задач прогнозування фінансових котувань.

Аналіз об’єкта дослідження за темою «Аналіз і прогнозування котувань на фондовій біржі методами машинного навчання» передбачає комплексне вивчення даних фондового ринку та застосування методів машинного навчання для аналізу та прогнозування руху курсів акцій.

Почнемо з того, що об’єктом дослідження в цій темі є фондовий ринок і пов’язані з ним дані, включаючи історичні дані про ціни, обсяги торгів, основні показники компанії, настрої новин та інші відповідні змінні. Аналіз цього об’єкта спрямований на виявлення закономірностей, тенденцій і зв’язків у даних, які можуть надати розуміння поведінки цін на акції.

Зазвичай аналіз починається з попередньої обробки даних, під час якої необроблені дані очищаються, нормалізуються та перетворюються у відповідний формат для аналізу. Цей крок забезпечує точність і узгодженість даних, що є вирішальним для подальшого аналізу.

Далі для аналізу об’єкта дослідження застосовуються різні методи машинного навчання. Ці методи можуть включати, але не обмежуватися регресійними моделями, алгоритмами класифікації, методами кластеризації та аналізом часових рядів. Регресійні моделі, такі як лінійна регресія або опорна векторна регресія, можна використовувати для моделювання зв’язку між вхідними змінними (наприклад, історичними курсами акцій, обсягами торгів) і цільовою змінною (наприклад, майбутніми курсами акцій). Алгоритми класифікації, такі як випадкові ліси або нейронні мережі, можна використовувати для прогнозування напрямку руху цін на акції (наприклад, вгору чи вниз). Методи кластеризації можуть бути використані для виявлення подібних шаблонів або груп у даних.

Методи аналізу часових рядів, такі як авторегресійна інтегрована ковзна середня (ARIMA) або мережі довгострокової короткочасної пам'яті (LSTM), можуть фіксувати часові залежності в даних про курс акцій для цілей прогнозування.

Під час аналізу використовуються різні показники ефективності та методи оцінки, щоб оцінити точність і ефективність моделей машинного навчання. Ці показники можуть включати середню квадратичну помилку (MSE), середню абсолютну помилку (MAE), точність, точність, відкликання або оцінку F1. Методи перехресної перевірки, такі як k-кратна перехресна перевірка, також можуть бути використані для оцінки продуктивності моделей на невидимих даних і пом'якшення переобладнання.

Аналіз об'єкта дослідження може включати в себе вибір ознак або техніку розробки ознак, щоб ідентифікувати найбільш релевантні змінні або перетворити вихідні дані в більш значущі представлення. Крім того, він може включати пошуковий аналіз даних (EDA), щоб отримати уявлення про розподіл даних, виявити викиди або виявити важливі зв'язки між змінними.

Крім того, аспект прогнозування аналізу зосереджений на передбаченні майбутніх коливань цін на акції на основі історичних даних і розроблених моделей машинного навчання. Це може допомогти інвесторам, трейдерам або фінансовим установам приймати обґрунтовані рішення та розробляти торгові стратегії.

Прогнозування — це процес прогнозування на основі даних минулого та теперішнього часу та, найчастіше, на основі аналізу тенденцій. Звичайним прикладом може бути оцінка певної змінної, що представляє інтерес, на певну дату в майбутньому. Передбачення – подібний, але більш загальний термін. Обидва можуть посилатися на формальні статистичні методи, що використовують часові ряди, дані поперечного зрізу чи поздовжніх даних, або, як альтернатива, менш формальні методи оцінювання. Використання може відрізнятися в різних сферах

застосування: наприклад, у гідрології терміни «прогноз» і «прогноз» іноді зарезервовані для оцінок значень у певний конкретний майбутній час, тоді як термін «прогноз» використовується для більш загальних оцінок, таких як кількість повеней, які відбуватимуться протягом тривалого періоду.[1]

Існуючі наразі методи прогнозування, на мою думку, можна розділити на наступні категорії:

Якісні методи прогнозування є суб'єктивними, заснованими на думці та судженнях споживачів та експертів; вони доречні, коли минулі дані недоступні. Зазвичай вони застосовуються до проміжних або довготривалих рішень. Прикладами якісних методів прогнозування є обґрунтована думка та судження [2], метод Дельфі, дослідження ринку та аналогія історичного життєвого циклу.

Для прогнозування майбутніх даних як функції минулих даних використовуються *моделі кількісного прогнозування*. Їх доцільно використовувати, коли доступні минулі числові дані та коли є розумне припущення, що деякі закономірності в даних, як очікується, збережуться і в майбутньому. Ці методи зазвичай застосовуються для прийняття рішень на короткому чи середньому діапазоні. Прикладами кількісних методів прогнозування є попит за останній період, прості та зважені ковзні середні за N-період, просте експоненціальне згладжування, прогнозування на основі моделі Пуассона [3] та мультиплікаційні сезонні індекси. Попередні дослідження показують, що різні методи можуть призвести до різного рівня точності прогнозування. Наприклад, виявлено, що нейронна мережа GMDH має кращу продуктивність прогнозування, ніж класичні алгоритми прогнозування, такі як Single Exponential Smooth, Double Exponential Smooth, ARIMA та нейронна мережа зворотного поширення.[4]

Методи усереднення. У цьому підході передбачення всіх майбутніх значень дорівнюють середньому за минулі дані. Цей підхід можна використовувати з будь-якими даними, де доступні попередні дані. Основною характеристикою методу середніх є те, що він формує прогноз на певний період часу шляхом усереднення

спостережуваних значень даних (тобто фактичних значень залежної змінної) за останні n періодів часу. [5]

Метод наївного прогнозування та його варіації. У цих методах використовується припущення, що продажі в наступному періоді будуть відповідати продажу в попередньому періоді. Наївні прогнози є найефективнішою моделлю прогнозування та є еталоном, з яким можна порівнювати більш складні моделі. Цей метод прогнозування підходить лише для даних часових рядів.[5] Використовуючи наївний підхід, формуються прогнози, які дорівнюють останньому спостережуваному значенню. Цей метод досить добре працює для економічних і фінансових часових рядів, які часто мають закономірності, які важко надійно і точно передбачити.[5] Якщо вважається, що часові ряди мають сезонність, сезонний наївний підхід може бути більш прийнятним, коли прогнози дорівнюють значенню минулого сезону.

До методів наївного прогнозування можна також віднести метод дрейфу, який полягає в тому, щоб дозволити прогнозам збільшуватися або зменшуватися з часом, коли величина змін за час (називається дрейфом) встановлюється як середня зміна, що спостерігається в історичних даних. Також до цієї категорії методів можна віднести і сезонний наївний метод, який враховує сезонність, встановлюючи кожен прогноз рівним останньому спостережуваному значенню того ж сезону. Наприклад, значення прогнозу для всіх наступних місяців квітня буде дорівнювати попередньому значенню, що спостерігалося в квітні.

Таким чином, аналіз об’єкта дослідження в темі «Аналіз та прогнозування котирувань на фондовому ринку з використанням методів машинного навчання» передбачає ретельне вивчення даних фондового ринку, застосування методів машинного навчання для виявлення закономірностей і взаємозв’язків, а також розробка прогностичних моделей для прогнозування майбутнього руху цін на акції. Він включає попередню обробку даних, вибір моделі, оцінку ефективності,

розробку функцій і методи прогнозування, спрямовані на покращення розуміння та прийняття рішень на фондовому ринку.

Також аналіз об’єкта дослідження в темі «Аналіз та прогнозування котирувань на фондовому ринку з використанням методів машинного навчання», важливо відзначити ітераційний характер процесу. Аналіз часто передбачає багаторазові ітерації розробки, оцінки та вдосконалення моделі для підвищення точності та надійності прогнозів.

Нарешті, аналіз об’єкта дослідження має супроводжуватися суворими процедурами валідації та тестування. Тестування поза вибіркою, де моделі оцінюються на основі невидимих даних, допомагає оцінити їхні можливості узагальнення та захищає від надмірного підбору даних навчання.

Загалом аналіз об’єкта дослідження за темою «Аналіз та прогнозування котирувань на фондовому ринку з використанням методів машинного навчання» є складним та динамічним процесом. Він передбачає застосування різних методів машинного навчання, розробку функцій, аналіз настроїв і оцінку ефективності для отримання інформації, розробки прогнозних моделей і прогнозування руху цін на акції. Сувора перевірка, врахування припущень і обмежень, а також врахування зовнішніх факторів є вирішальними для отримання надійних і практичних результатів у цій галузі досліджень.

Важливо визнати невід’ємну невизначеність, пов’язану з прогнозуванням фондового ринку. На фінансові ринки впливає безліч факторів, багато з яких непередбачувані. Тому, хоча методи машинного навчання пропонують цінні інструменти для аналізу та прогнозування, їх слід використовувати як інструменти підтримки прийняття рішень, а не як абсолютні прогнози. Дуже важливо інтерпретувати результати в поєднанні з експертними знаннями в галузі, знаннями ринку та іншою відповідною інформацією.

Таким чином, аналіз об’єкта дослідження в темі «Аналіз та прогнозування котирувань на фондовому ринку з використанням методів машинного навчання» вимагає ретельного врахування якості даних, наявності релевантних ознак, інтерпретованості моделі та ітераційного підходу. Звертаючись до цих факторів, використовуючи методи ансамблю та визнаючи притаманну фондовому ринку невизначеність, аналіз може надати цінну інформацію та сприяти прийняттю більш обґрунтованих рішень у сфері аналізу та прогнозування фондового ринку.

1.2. Методи машинного навчання для прикладних задач прогнозування фінансових котирувань

Однією з ключових переваг методів машинного навчання в прогнозуванні фондового ринку є їх здатність адаптуватися та вчитися на нових даних. Ці моделі можна навчати на великих історичних наборах даних, а потім оновлювати новими даними, коли вони стають доступними, що дозволяє їм фіксувати мінливу ринкову динаміку та відповідним чином адаптувати свої прогнози. Ця адаптивність особливо цінна на динамічних і нестабільних ринках, де традиційним моделям прогнозування може бути важко встигати за мінливими тенденціями та моделями.

Ефективний аналіз великих наборів даних за допомогою машинного навчання та технологій штучного інтелекту дає організаціям конкурентні переваги, отримуючи уявлення про поведінку клієнтів, ефективність процесів, вплив на бізнес та багато іншого. Ці переваги є вирішальними для досягнення успіху в бізнес-середовищі, що швидко змінюється [7].

Методи машинного навчання більш вимогливі до обчислень, ніж статистичні, потребують більше ресурсів. Однак у бізнес-додатках із величезними обсягами даних методи машинного навчання можуть краще підходити для прогнозів через велику кількість залучених функцій даних і той факт, що використовуваний алгоритм може бути не лінійним або простим.

На таблиці 1.1 ми бачимо, що методи машинного навчання працюють з більшою точністю, ніж традиційні методи. Дослідження було проведено розробниками компанії Genpact. Цей результат був орієнтовним на основі показників середньої абсолютної відсоткової помилки, де прогноз методом машинного навчання мав нижче значення, що показує вищу точність у прогнозі.[6]

Таблиця 1.1

Порівняння результатів роботи МН до традиційного методу

Метрика помилок	МН	Традиційні методи
Середня абсолютна відсоткова помилка	11.61%	15.17%
Середньо-квадратична похибка	529.662	738.225
Зважена середній абсолютний відсоток помил.	11.68%	15.26%
Зважена середньо-квадратична похибка	539.633	749.471

Підсумовуючи, методи прогнозування МН є високоефективними в програмах, метою яких є навчання на наборах даних, які мають багато функцій, а пояснення моделі не настільки критичне. Для випадків використання з великими наборами даних методи прогнозування ML, здається, працюють з вищою точністю, але також мають більші обчислювальні вимоги та не так пояснюються. Традиційні статистичні методи можуть працювати краще в одновимірних додатках, метою яких є аналіз та узагальнення даних. Цей тип даних має менше унікальних функцій, а прозорість і пояснення моделі дуже потрібні.

У XX столітті, з розвитком техніки та появою нових відкритих даних, збільшилася доступність проведення аналізу – інформацію стало простіше зберігати та обробляти. Численні методи, що використовуються в статистиці та математиці, стали застосовуватися до фінансових даних. Ранні дослідження прогнозу ціни датуються 1960-1970 роками: роботою Eugene Fama "Поведінка

ринкової ціни на біржі", і роботою "Теорія випадкових блукань", яка була розвинена Cootner у 1964, Fama та Fisher у 1969 році. Ці ранні моделі припускають, що ринкова ціна може бути спрогнозована з більш ніж 50% точністю. Згодом стали з'являтися дослідження (див. Malkiel, 2003; Smith, 2003; Bollen, Mao & Zeng, 2011), які підтверджують, що ринок може бути спрогнозований. Точність попередніх досліджень щодо прогнозування вартості не перевищувала 83% (див. Lawrence, 1997; Vu, Chang, Ha & Collier, 2012). Ефективної системи немає і багато, пов'язані з цією темою, обговорення стосуються підвищення точності прогнозування. Доступність інвестування та потенційні обсяги прибутку роблять завдання прогнозування ціни фінансових активів актуальною.

Ціна акцій компанії на фондовому ринку залежить від багатьох факторів. Чинники, що впливають на ціноутворення фондових активів, залежно від методу аналізу, що застосовується, можуть варіюватися від макроекономічних показників до технічних індикаторів, таких як ковзна середня та інші. Вибір факторів, що впливають на ціноутворення залежить від фінансового інструменту, а також низки припущень, присутніх у моделі. Наприклад, ціни на облігації державної позики, згідно з дослідженням Річарда Колмена (1989) багато в чому залежать від макроекономічних показників, тоді як ціни на високоліквідні ф'ючерси прогнозуються шляхом виділення високочастотних технічних індикаторів, таких як ціни за попередній період та тощо.

Досягнення точного прогнозу фондового ринку сильно впливає на інвесторів, надаючи їм інструменти для прийняття кращих рішень на основі даних. Такий прогноз може підвищити прибутковість їхніх інвестицій, допомогти їм вибрати найбільш прибуткові акції та зменшити інвестиційний ризик. Крім того, покращення інструментів для прогнозування фондового ринку може допомогти біржовим трейдерам використовувати кращу інформацію, як-от історичні ціни акцій та новини. [7]

Наукова спільнота досліджувала різні способи прогнозування фондового ринку [7]. В основному, існує два способи підійти до прогнозу фондового ринку: фундаментальний аналіз, коли основні фактори, що впливають на компанії або галузі, використовуються як прогностичні атрибути; і технічний аналіз, де прогнозними атрибутами є переважно історичні ціни та обсяги.

Як видно з таблиці 1.1 методи машинного навчання значно обходять традиційні методи прогнозування. Але в той самий час постає задача визначення, який саме підхід (метод) машинного краще обрати. Звичайно увагу необхідно акцентувати на методах, які розроблені та пристосовані для задач прогнозування, зокрема до задачі прогнозування часових рядів. З існуючих методів для порівняння були відібрані метод штучних нейронних мереж та метод опорних векторів.

У машинному навчанні метод опорних векторів — це контрольовані моделі навчання з відповідними алгоритмами навчання, які аналізують дані для класифікації та регресійного аналізу. Розроблені в AT&T Bell Laboratories Володимиром Вапником з колегами. Цей алгоритм можна використовувати для прогнозування цін закриття наступного дня за допомогою SVR (Support Vector Regression), прогнозування руху вгору/вниз за допомогою SVM для класифікації або прогнозування інтервалу ціни. Версія SVM для регресії була запропонована в 1996 році Володимиром Н. Вапником, Харрісом Друкером, Крістофером Дж. К. Берджесом, Ліндою Кауфман і Олександром Дж. Смолою. Цей метод називається регресією опорного вектора (SVR).[8][9]

Штучна нейронна мережа – це система, що складається з багатьох простих обчислювальних елементів (нейронів), певним чином пов'язаних між собою. Найбільш поширеними є багатошарові мережі, в яких нейрони об'єднані в шари. Штучні нейронні мережі, безумовно, є найбільш дослідженим методом прогнозування фондового ринку. Це сімейство методів дозволяє вивчати нелінійність поведінки фондового ринку, майже без спеціальних припущень.

Методи нейронних мереж використовуються як вхідні дані переважно часових рядів і технічних індикаторів. Вони використовувалися для прогнозування цін закриття наступного дня, а також для прогнозування руху вгору вниз.[9]

Згідно до проведених у 2013 році досліджень[10], які біли присвячені порівнянню методу опорних векторів з методом нейронних мереж для передбачення індексів акцій, штучна нейронна мережа поступається в точності методу опорних векторів. «У цьому дослідженні було зроблено висновок, що SVM є багатообіцяючою альтернативою прогнозуванню часових рядів, оскільки забезпечує менший середньо-квадратичну похибку» .

У таблиці 1.2 наведені результати цих досліджень по показнику середньо-квадратичної похибки. Можна побачити, що відмінність не є разючою.

Таблиця 1.2

Результати порівняння методу опорних векторів з моделлю нейронних мереж

Рік	Тренувальні дані		Дані валідації	
	МОВ	ШНМ	МОВ	ШНМ
2008	1.1292923	1.2145074	1.1292923	1.1345074
2009	0.60363	0.7657106	0.60363	0.757106
2010	0.1908253	0.2125603	0.1908253	0.2525603
2011	0.2727183	0.3696291	0.2727183	0.3696291
2012	0.1001534	0.164633	0.1001534	0.164633

Але якщо звернутися до роботи[9] 2017 року «Порівняння між SVM та багатошаровим перцептроном у прогнозуванні фінансового ринку, що розвивається: колумбійський фондовий ринок» то можна побачити, що штучна нейронна мережа, яка представлена багатошаровим перцептроном демонструє

кращі результати. Див. Додаток А. Саме тому для подальшої роботи було обрано метод штучних нейронних мереж, а саме модель багатошарового перцептрону.

1.3. Постановка задачі аналізу і прогнозування котувань акцій на фондовому ринку методами машинного навчання

Завдання аналізу та прогнозування котирувань фондового ринку за допомогою нейромережевої моделі та методу опорних векторів передбачає розробку комплексної системи, яка може аналізувати та прогнозувати коливання цін на фондовому ринку за допомогою двох основних підходів: нейронних мереж та методів опорних векторів.

Мета аналізу полягає в тому, щоб зрозуміти залежності та зв’язки між вхідними змінними, такими як історичні біржові дані, обсяги торгів, фундаментальні показники тощо, і вихідною змінною – майбутніми котируваннями акцій. Це дозволяє системі зрозуміти всю складність ринку та визначити ключові фактори, що впливають на ціни акцій. Побічною метою буде дослідження і порівняння результатів роботи нейронної мережі з результатами роботи методу опорних векторів.

Для початку потрібно зібрати відповідний набір даних. Цей набір даних має включати історичні котирування акцій, обсяги торгів, фундаментальні дані про компанії, новини та інші важливі змінні. Ці дані слугуватимуть вхідними параметрами для моделей.

Далі нейронні мережі розробляються як потужні алгоритми машинного навчання, здатні розкривати складні залежності між вхідними даними та вихідними прогнозами. Можуть бути використані різні архітектури нейронних мереж, наприклад рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) або комбінації цих моделей. Мережі навчаються на історичних даних, де вхідними параметрами є історичні котирування та додаткові змінні, а вихідними є майбутні

біржові котирування. Процес навчання передбачає оптимізацію ваг і параметрів мережі для мінімізації похибки передбачення.

Використовуючи метод опорного вектора (SVM), модель може аналізувати дані та будувати межі рішень, які розділяють різні класи або прогнозують значення біржових котирувань. SVM може використовувати різні ядра (наприклад, лінійну, радіально-базисну функцію (RBF) або поліноміальну), які допомагають моделі знайти оптимальну межу рішення. SVM навчається з використанням історичних даних, де вхідними параметрами є вхідні змінні, а вихідними є майбутні котирування акцій.

Після навчання нейронної мережі та моделей SVM їх можна використовувати для прогнозування майбутніх котирувань акцій. Нові вхідні дані, включаючи історичні дані, а також нові значення змінних, такі як останні котирування чи новини, використовуються моделями для прогнозування майбутніх котирувань акцій.

Після отримання прогнозів може бути проведена оцінка їх точності та достовірності. Цього можна досягти шляхом порівняння прогнозованих значень з реальними даними про біржові котирування. Для оцінки точності моделей можна використовувати такі показники, як середня квадратична помилка (MSE), середня абсолютна помилка (MAE) або коефіцієнт детермінації (R^2).

Управління ризиками також є важливим аспектом такого завдання. Прогнозування біржових котирувань пов’язане зі значними фінансовими ризиками, тому ретельна оцінка потенційних ризиків і використання стратегій управління ризиками мають вирішальне значення для пом’якшення потенційних негативних наслідків.

Підсумовуючи, завдання аналізу та прогнозування котирувань фондового ринку за допомогою моделі нейронної мережі та методу опорних векторів

передбачає розробку комплексної системи, яка включає передові методи машинного навчання. Ця система спрямована на аналіз складності та нелінійності фондового ринку, визначення ключових факторів, що впливають на ціни акцій, і створення точних прогнозів щодо майбутніх котирувань акцій.

1.4 Аналіз актуальності застосування методів машинного навчання для прогнозування котувань акцій на фондовому ринку.

На основі розглянутої інформації ми можемо стверджувати, що прогнозування фінансових котувань є актуальною та досліджуваною темою. Це передусім пов’язано з потенційно великими доходами у сфері. Існує попит на прогнози на фінансовому ринку. Така попит приводить до того, що задачу вирішують різними методами, з’являються продукти націлені на прогнозування, сфера стрімко розвивається.

Використання методів машинного навчання для прогнозування котирувань фондового ринку привернуло значну увагу в останні роки.[11] Технології машинного навчання дозволяють аналізувати величезну кількість історичних даних, виявляти складні закономірності та зв’язки та робити точні прогнози. У контексті прогнозування фондового ринку моделі машинного навчання можуть вивчати історичні дані про ціни, обсяги торгів, технічні індикатори, настрої новин та інші відповідні фактори для створення прогнозів майбутніх котирувань на фондовому ринку.

Однією з ключових переваг методів машинного навчання є їх здатність фіксувати нелінійні зв’язки та адаптуватися до мінливих умов ринку. Традиційні економетричні моделі часто припускають лінійні зв’язки, і їм може бути важко охопити складність і динамічний характер фінансових ринків. Моделі машинного навчання, з іншого боку, можуть впоратися з нелінійністю та включати широкий спектр вхідних змінних для охоплення різноманітної динаміки ринку.

Також одним з важливих аспектів використання методів машинного навчання для прогнозування фондового ринку є розробка функцій. Розробка функцій передбачає вибір і перетворення вхідних змінних для підвищення передбачуваної потужності моделі. Цей процес може включати вилучення технічних індикаторів (таких як ковзні середні, індекс відносної сили або смуги Боллінджера[14]) із необроблених цінових даних, включення макроекономічних показників, аналіз настроїв у новинних статтях або даних із соціальних мереж, а також урахування факторів, характерних для ринку. Ефективна розробка функцій може значно вплинути на продуктивність моделей машинного навчання, надаючи їм релевантні та інформативні функції введення.

Існує кілька популярних алгоритмів машинного навчання, які використовуються для прогнозування фондового ринку.[12] Одним із широко поширених підходів є використання рекурентних нейронних мереж (RNN) та їх варіантів, мереж довготривалої короткочасної пам’яті (LSTM). RNN розроблені для обробки послідовних даних, що робить їх добре придатними для завдань прогнозування часових рядів. Мережі LSTM, завдяки своїй здатності зберігати інформацію протягом більших інтервалів часу, можуть фіксувати залежності та довгострокові шаблони в даних. Ці моделі показали багатообіцяючі результати в охопленні тимчасової динаміки та створенні точних прогнозів у прогнозуванні фондового ринку.

В останні роки методи глибокого навчання, зокрема згорткові нейронні мережі (CNN), були застосовані для прогнозування фондового ринку. CNN чудово вилучають функції з вхідних даних, таких як діаграми цін на акції чи новинні статті, за допомогою згорткових фільтрів. Вивчаючи ієрархічні представлення даних, моделі CNN можуть розкривати значущі моделі та зв’язки, які сприяють прогнозуванню фондового ринку.

Метод опорних векторів (SVM) також широко використовуються в прогнозуванні фондового ринку. SVM мають на меті знайти оптимальну гіперплощину, яка розділяє точки даних різних класів. У контексті прогнозування фондового ринку SVM можуть вивчати історичні дані, щоб установити межі рішень, які розділяють періоди зростання або зниження цін на акції.[13] Використовуючи різні функції ядра, такі як лінійна, радіальна базисна функція (RBF) або поліноміальна, SVM можуть фіксувати складні зв'язки між вхідними змінними та рухами фондового ринку.

Мінг-Чі Лі використав метод опорних векторів (SVM) разом із гібридним методом вибору ознак прогнозування тенденцій ринку акцій [15]. Набір даних у цьому дослідженні є набором даних індексу NASDAQ у базі даних Тайванського економічного журналу (TEJD) за 2009 рік. У частині відбору ознак використовувався гібридний метод, у ролі обгортки виступав підтримуваний послідовний прямий пошук (SSFS). Ще одна перевага даної роботи полягає у тому, що в ній була розроблена докладна процедура налаштування параметрів із зазначенням продуктивності при різних значеннях параметрів. Чітка структура моделі відбору ознак є також евристикою для первинного етапу структурування моделі. Одним з обмежень було те, що продуктивність SVM порівнювалася лише з нейронною мережею із зворотним розповсюдженням (BPNN) і не порівнювалася з іншими алгоритмами машинного навчання.

Sirignano та Cont [16] використали рішення глибокого навчання, навчене на універсальному наборі ознак фінансових ринків. Набір даних, що використовується, включав записи всіх угод на купівлю та продаж, а також скасування ордерів для приблизно 1000 акцій NASDAQ через книгу ордерів 25 біржі. NN складається з трьох шарів з блоками LSTM і шару feed-forward з випрямленими лінійними блоками (ReLU) в кінці зі стохастичним алгоритмом градієнтного спуску (SGD) в якості оптимізації. Їх універсальна модель була здатна

узагальнювати та охоплювати акції, відмінні від тих, що були у навчальних даних. Хоча вони відзначили переваги універсальної моделі, вартість навчання все ще була дорогою. Тим часом через неявне програмування алгоритму глибокого навчання неясно, чи є марні ознаки, забруднені при подачі даних у модель. Автори виявили, що було б краще, якби вони виконали частину відбору ознак перед навчанням моделі, і вважають це ефективним способом зниження обчислювальної складності.

Для оцінки продуктивності моделей машинного навчання зазвичай використовуються різні показники, включаючи середню квадратичну помилку (MSE), середню абсолютну помилку (MAE), середньоквадратичну помилку (RMSE) і точність направлення. Техніки ретроспективного тестування та перехресної перевірки часто використовуються для оцінки передбачуваної потужності моделей і забезпечення їх можливостей узагальнення.

Підсумовуючи, використання методів машинного навчання для прогнозування котирувань фондового ринку відкриває багатообіцяючі можливості використовувати величезні обсяги даних, фіксувати складні закономірності та робити точні прогнози. Такі методи, як RNN, мережі LSTM, випадкові ліси, SVM і CNN, продемонстрували свою ефективність у моделюванні динаміки фінансових ринків. Однак ретельний розгляд якості даних, вибору функцій, архітектури моделі та управління ризиками має важливе значення для успішного впровадження. Постійні дослідження та вдосконалення алгоритмів машинного навчання, а також вдосконалення методів збору та попередньої обробки даних сприяють постійному розвитку та вдосконаленню моделей прогнозування фондового ринку.

Підводячи підсумок, можна сказати, що використання методів машинного навчання для прогнозування фондового ринку пропонує великий потенціал для виявлення складних закономірностей, отримання цінної інформації з великих наборів даних і створення точних прогнозів. Такі методи, як RNN, мережі LSTM, випадкові ліси, SVM і CNN, разом із розробкою функцій і методами ансамблю,

сприяють розвитку прогнозних моделей на фінансових ринках. Однак для ефективного застосування машинного навчання для прогнозування фондових ринків необхідний ретельний розгляд якості даних, інтерпретації моделі, управління ризиками та природи фінансових ринків, що постійно розвивається. Постійні дослідження та розробки в цій галузі сприяють постійному вдосконаленню моделей прогнозування та дослідженню нових методів для більш точних і надійних прогнозів.

РОЗДІЛ 2. АЛГОРИТМИ МАШИННОГО НАВЧАННЯ ТА СПОСОБИ ЇХ НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ КОТУВАНЬ НА ФОНДОВІЙ БІРЖІ

2.1. Математична модель штучних нейронних мереж

Штучна нейронна мережа – це система, що складається з багатьох простих обчислювальних елементів (нейронів), певним чином пов'язаних між собою. Найбільш поширеними є багатошарові мережі, в яких нейрони об'єднані в шари. Шар – це сукупність нейронів, на які в кожний такт часу паралельно надходить інформація від інших нейронів мережі, тобто виходи нейронів з'єднуються з входами інших нейронів, так сигнал від одного елемента передається іншим.

Після того, як визначено кількість шарів і число елементів в кожному з них, мережу потрібно навчити [10][17], тобто визначити значення для ваг і порогів мережі, які мінімізували б помилку прогнозу, що видається мережею. Помилка для конкретної конфігурації мережі визначається шляхом прогону через мережу всіх наявних спостережень і порівняння вихідних значень, що реально видаються, із бажаними (цільовими) значеннями.

Типовий приклад роботи нейронної мережі показаний на рисунку 2.1. Вхідний шар призначений просто для введення значень вхідних змінних. Кожен з прихованих і вихідних нейронів з'єднаний з усіма елементами попереднього шару.

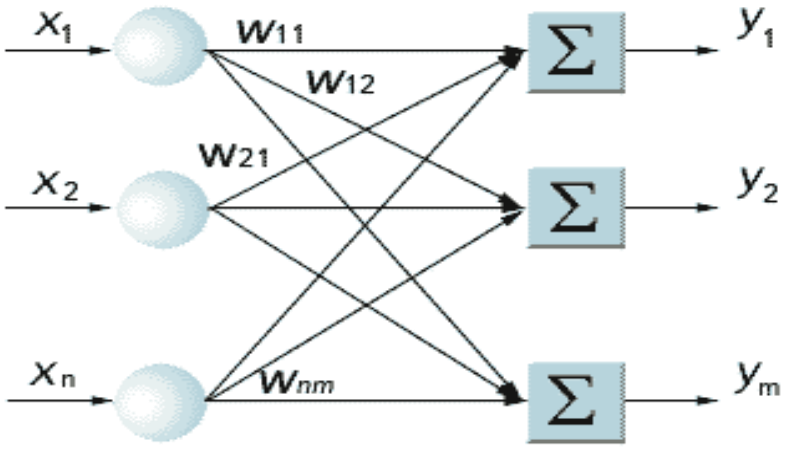


Рисунок 2.1. – Схема одношарової нейронної мережі

На сьогоднішній момент існує кілька десятків структур нейронних мереж. Оскільки всі штучні нейронні мережі базуються на концепції нейронів, з'єднань та передатних функцій, існує подібність між різними структурами нейронних мереж. Для процесу навчання необхідно мати модель зовнішнього середовища, у якій функціонує нейрона мережа – потрібну для вирішення задачі інформацію. Також, необхідно визначити, як модифікувати вагові параметри мережі. Алгоритм навчання означає процедуру, у якій використовуються правила навчання для налаштування ваг [18].

Навчити нейронну мережу – значить, повідомити їй, чого ми від неї домагаємося. Нейронна мережа може навчатися з вчителем або без нього. Після багаторазового пред'явлення прикладів ваги нейронної мережі стабілізуються, причому нейронна мережа дає правильні відповіді на всі (або майже всі) приклади з бази даних. Детальніше про це далі у пункті 2.1.7.

Однією з досить значимих областей застосування нейронних мереж у фінансовій сфері є прогнозування на фондовому ринку. Застосування нейронних мереж є досить потужним методом прогнозування, який дозволяє відтворювати досить складні залежності. Нейронні мережі для прогнозування фондового ринку мають наступний перелік переваг:

- Простота у використанні, так як нейронні мережі навчаються на прикладах.
- Нейронні мережі привабливі з інтуїтивної точки зору, тому що вони засновані на примітивній біологічній моделі нервових систем[19].
- Передбачення фінансових часових рядів - необхідний елемент будь-якої інвестиційної діяльності.

2.1.1. Штучний нейрон

На вхід штучного нейрона поступає деяка множина сигналів, кожний з яких є виходом іншого нейрона. Кожний вхід перемножується з відповідною вагою,

аналогічної синаптичній силі, і всі доданки підсумовуються, визначаючи рівень активації нейрона.

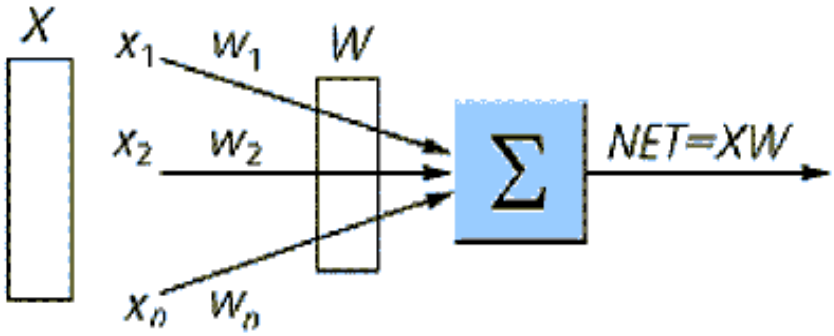


Рисунок 2.2 - Штучний нейрон

На рисунку 2.2 представлена ця модель. Якщо презентувати вищенаведений рисунок математичною формулою, то отримаємо:

$$y_k = \varphi(\sum_{j=0}^{n} \omega_{kj}\, x_j) \qquad (2.1)$$

де φ - Передавальна функція (зазвичай є пороговою функцією)

k – деякий заданий нейрон,

n+1 – кількість вхідних сигналів,

$x_1, x_2, \ldots x_n$ – множина вхідних сигналів,

$w_{k1}, w_{k2}, \ldots w_{kn}$ – ваги.

Тут позначених $x_1, x_2, \ldots x_n$, надходить на штучний нейрон. Кожний сигнал перемножується з відповідною вагою $w_1, w_2, \ldots w_n$, і надходить на підсумовуючий блок, позначений S. Кожна вага відповідає "силі" одного біологічного синаптичного зв'язку. Множина ваг в сукупності позначається вектором W. Підсумовуючий блок, складає зважені входи алгебраїчно, створюючи вихід, який ми будемо називати NET. У векторних позначеннях це може бути коротко записане таким чином: NET = XW.

2.1.2. Активаційна функція

У штучних нейронних мережах функція активації вузла визначає вихід цього вузла на вхід або набір входів. Стандартну інтегральну схему можна розглядати як цифрову мережу функцій активації, яка може бути «ВКЛЮЧЕНО» (1) або «ВИМКНЕНО» (0), залежно від входу.

Сигнал NET далі, як правило, перетворюється активаційною функцією F і дає вихідний нейронний сигнал OUT. Активаційна функція може бути звичайною лінійною функцією OUT = K(NET), де K константа порогової функції; OUT = 1, якщо NET > T, OUT = 0 в інших випадках, де T деяка постійна порогова величина, або ж функція, що точніше моделює нелінійну передатну характеристику біологічного нейрона і надає нейронній мережі великі можливості. Це можна відобразити формулою:

$$\sigma(\mathrm{x}) = \frac{1}{(1 + \mathrm{e}^{-x})} \tag{2.2}$$

де x – це вхідний сигнал

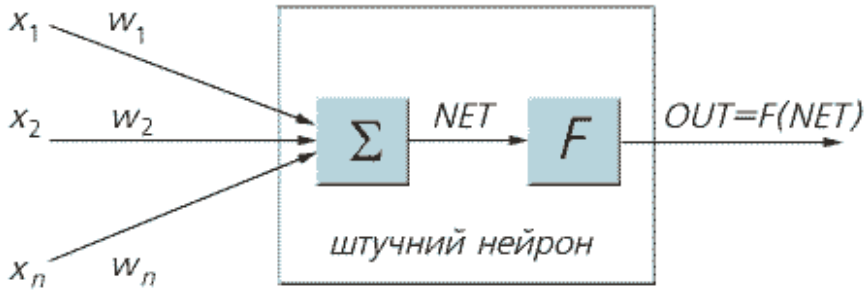


Рисунок 2.3 – Штучний нейрон з активаційною функцією

На рисунку 2.3 блок, позначений F, приймає сигнал NET і видає сигнал OUT. Якщо блок F звужує діапазон зміни величини NET так, що при будь-яких значеннях NET значення OUT належать деякому кінцевому інтервалу, то F називається "стискаючою" функцією. В якості "стискаючої" функції часто використовується логістична або "сигмоїдальна" функція. Ця функція математично виражається як

Активаційну функцію можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Коефіцієнт посилення обчислюється як

відношення приросту величини OUT до невеликого приросту величини, що викликає NET. Слабкі сигнали потребують великого мережевого посилення, щоб дати придатний до використання вихідний сигнал. Однак підсилювальні каскади з великими коефіцієнтами посилення можуть привести до насичення виходу шумами підсилювачів. Сильні вхідні сигнали в свою чергу також будуть приводити до насичення підсилювальних каскадів, виключаючи можливість корисного використання виходу. Центральна область логістичної функції, що має великий коефіцієнт посилення, вирішує проблему обробки слабих сигналів, в той час як області з падаючим посиленням на позитивному і негативному кінцях підходять для великих збуджень. Таким чином, нейрон функціонує з великим посиленням в широкому діапазоні рівня вхідного сигналу.

Рисунок 2.4 – Сигмоїдальна логістична функція

Іншою активаційною функцією, що широко використовується є гіперболічний тангенс. За формою вона схожа з логістичною функцією і часто використовується біологами як математична модель активації нервової клітки. Як активаційна функція штучної нейронної мережі вона записується таким чином:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.3}$$

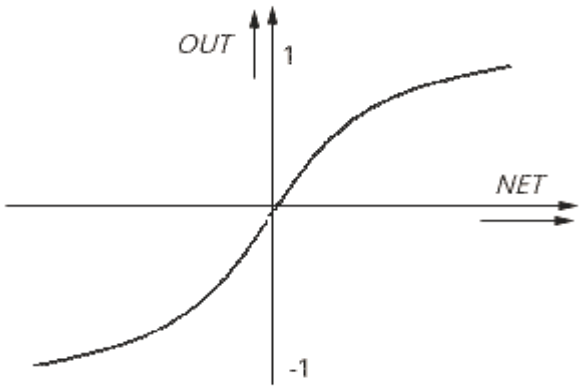


Рисунок 2.5 – Функція гіперболічного тангенса

Подібно логістичній функції гіперболічний тангенс є S-образною функцією, але він симетричний відносно початку координат, і в точці NET = 0 значення вихідного сигналу OUT дорівнює нулю (див. рисунок 2.5). На відміну від логістичної функції гіперболічний тангенс приймає значення різних знаків, що виявляється вигідним для ряду мереж.

2.1.3. Багатошарові штучні нейронні мережі

Більш великі і складні нейронні мережі мають, як правило, і великі обчислювальні можливості. Виявилося, що такі багатошарові мережі володіють більшими можливостями, ніж одношарові, і в останні роки були розроблені алгоритми для їх навчання.[22]

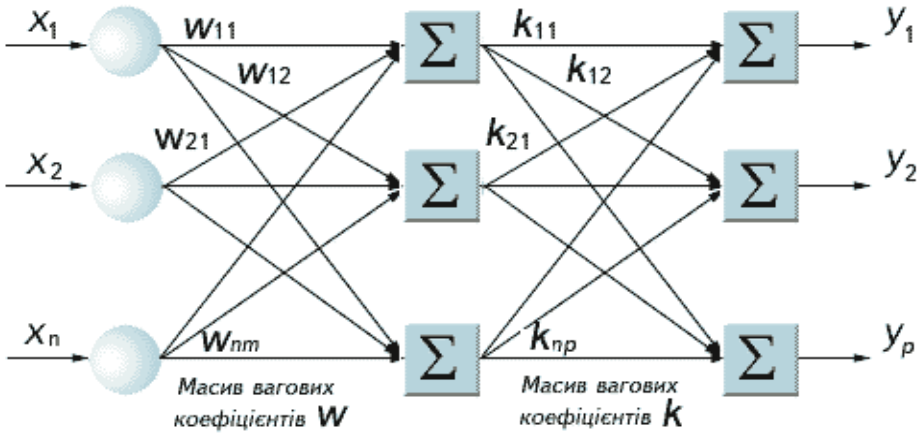


Рисунок 2.6. – Двошарова нейронна мережа

Багатошарові мережі можуть утворюватися каскадами прошарків. Вихід одного прошарку є входом для подальшого прошарку. Подібна мережа показана на рисунку 2.6. і знов зображена з всіма з'єднаннями.

2.1.4. Нелінійна активаційна функція

Багатошарові мережі не можуть привести до збільшення обчислювальної потужності в порівнянні з одношаровою мережею лише в тому випадку, якщо активаційна функція між прошарками буде нелінійною. Обчислення виходу прошарку полягає в множенні вхідного вектора на першу вагову матрицю з подальшим множенням результуючого вектора на другу вагову матрицю.

$$(XW_1)W_2$$

Оскільки множення матриць асоціативне, то

$$X(W_1\ W_2).$$

Це показує, що двошарова лінійна мережа еквівалентна одному прошарку з ваговою матрицею, рівною виробленню двох вагових матриць. Отже, будь-яка багатошарова лінійна мережа може бути замінена еквівалентною одношаровою мережею. Таким чином, для розширення можливостей мереж в порівнянні з одношаровою мережею необхідна нелінійна активаційна функція.

2.1.5. Навчання штучних нейронних мереж

Мережа навчається, щоб для деякої множини входів давати бажану множину виходів. Кожна така вхідна множина розглядається як вектор. Навчання здійснюється шляхом послідовного пред'явлення вхідних векторів з одночасним налаштуванням ваг відповідно до алгоритму зворотного поширення. Зворотне поширення — це метод обчислення градієнту функції втрат по відношенню до ваг в ШНМ. Уточнення ваг зворотного поширення можливо здійснювати за допомогою стохастичного градієнтного спуску із застосуванням наступного рівняння:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}} + \xi(t) \qquad (2.4)$$

2.1.6. Стохастичний градієнтний спуск

Стохастичний градієнтний спуск (часто скорочено SGD) — це ітераційний метод оптимізації цільової функції з відповідними властивостями гладкості. Його можна розглядати як стохастичне наближення оптимізації градієнтного спуску, оскільки воно замінює фактичний градієнт (розрахований з усього набору даних) його оцінкою (розрахованою з випадково вибраної підмножини даних).

І статистичне оцінювання, і машинне навчання розглядають проблему мінімізації цільової функції, яка має вигляд суми:

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w) \tag{2.5}$$

2.1.7. Середньоквадратична похибка як функція втрат

СКП вимірює усереднення квадратів похибок — тобто, середнє квадратичної різниці між оцінками значень та справжнім значенням. Якщо вектор з n передбачень породжується з вибірки n точок даних на всіх змінних, Y є вектором спостережуваних значень передбачуваної змінної, а $\hat{Y}$ є передбаченими значеннями (наприклад, як із допасовування найменшими квадратами), тоді СКП цього передбачувача в межах цієї вибірки обчислюється як:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.6}$$

Математичні переваги середньоквадратичної похибки особливо очевидні при її використанні для аналізу продуктивності лінійної регресії, оскільки це дозволяє розділити дисперсію в наборі даних на дисперсію, що пояснюється моделлю, та дисперсію, що пояснюється випадковістю.

2.1.8. Навчання без вчителя

Концепція навчання без вчителя розвинена Кохоненом [21] і багатьма іншими науковцями, воно не потребує цільового вектора для виходів і, отже, не вимагає

порівняння з встановленими ідеальними відповідями. Навчальна множина складається лише з вхідних векторів. Навчальний алгоритм налаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення досить близьких вхідних векторів давало однакові виходи. Процес навчання, виділяє статистичні властивості навчальної множини і групує схожі вектори в класи. Пред'явлення на вхід вектора з даного класу дасть певний вихідний вектор, але до навчання неможливо передбачити, який вихід буде вироблятися даним класом вхідних векторів.

2.1.9. Алгоритми навчання

У штучній нейронній мережі, що використовує навчання за Хебом [20], нарощування ваги визначається виробленням рівнів збудження передаючого і приймаючого нейронів. Це можна записати як

$$w_{ij}(n+1) = w(n) + \alpha\, OUT_{\,i}\, OUT_j, \qquad (2.7)$$

де $w_{ij}(n)$ – значення ваги від нейрона і до нейрона j до налаштування,

$w_{ij}(n+1)$ - значення ваги від нейрона *i* до нейрона *j* після налаштування,

α – коефіцієнт швидкості навчання,

$OUT_{\,i}$ – вихід нейрона *i* та вхід нейрона *j*,

OUT_j – вихід нейрона *j*.

2.1.10. Недоліки штучних нейронних мереж і шляхи їх усунення.

Одним із недоліків штучних нейронних мереж (ШНМ) є їх схильність до переобладнання. Переобладнання відбувається, коли мережа вчиться добре працювати з навчальними даними, але не може узагальнити невидимі дані. Це може призвести до низької продуктивності та обмеженої застосовності моделі.

Іншою проблемою є потреба у великій кількості позначених навчальних даних. ШНМ зазвичай вимагають значної кількості позначених даних для ефективного навчання та створення точних прогнозів. Отримання та позначення

таких даних може бути трудомістким і дорогим, особливо в доменах з обмеженою кількістю доступних позначених даних.

Крім того, навчання ШНМ може бути обчислювально дорогим і трудомістким, особливо для глибоких нейронних мереж із численними рівнями та параметрами. Процес навчання часто вимагає значних обчислювальних ресурсів, включаючи потужне обладнання та значний час навчання.

Щоб усунути ці недоліки та покращити ШНМ, можна розглянути кілька підходів:

Техніки регулярізації: методи регулярізації, такі як регулярізація L1 і L2, випадання та рання зупинка, можуть допомогти запобігти переобладнанню. Ці методи додають обмеження процесу навчання мережі, зменшуючи його залежність від конкретних шаблонів у навчальних даних.

Розширення даних: методи розширення даних включають штучне розширення навчального набору даних шляхом застосування перетворень або додавання шуму до існуючих даних. Такий підхід збільшує різноманітність навчальних прикладів і допомагає мережі краще узагальнювати.

Передача навчання: передача навчання використовує попередньо підготовлені моделі на великих наборах даних і налаштовує їх для конкретних завдань з обмеженими позначеними даними. Такий підхід дає змогу моделі використовувати знання, отримані з пов’язаного завдання чи домену, зменшуючи потребу у великих позначених даних.

Покращення архітектури: Дослідники постійно досліджують нові архітектури та модифікації мереж, щоб підвищити продуктивність і ефективність ШНМ. Такі методи, як згорточні нейронні мережі (CNN), рекурентні нейронні мережі (RNN) і механізми уваги, показали покращення в окремих областях.

Апаратне прискорення: використання спеціалізованого апаратного забезпечення, такого як графічні процесори (GPU) або тензорні процесори (TPU),

може значно прискорити процеси навчання та логічного висновку, зменшуючи обчислювальний тягар ШНМ.

Активне навчання: підходи до активного навчання спрямовані на оптимізацію процесу маркування даних шляхом вибору найбільш інформативних точок даних для маркування. Активно вибираючи зразки даних для позначення, модель може досягти кращої продуктивності з меншим позначеним набором даних.

Враховуючи ці підходи, розробники та дослідники можуть пом'якшити обмеження ШНМ та підвищити їхню ефективність у різних додатках.

2.2. Математична модель методу опорних векторів

Машина опорних векторів, або Метод опорних векторів – метод машинного навчання, який застосовується для вирішення задач класифікації. Даний метод базується на побудові оптимальної роздільної гіперплощини [1-3]. Метод був розроблений протягом 1960-70хх років, особливо сильно поширився протягом 90-х років XX-го століття. Навчання в самому методі зводиться до вирішення задачі квадратичного програмування, яке має єдине рішення.

Обчислення за допомогою такої задачі залишається досить ефективним навіть при вибірці в сотні тисяч об'єктів. Розв'язок має різні властивості, зокрема – розрідженості: положення вищезгаданої гіперплощини залежить від малої долі навчальних об'єктів. Саме ці об'єкти являються опорними векторами, завдяки яким метод отримав свою назву. За допомогою введення так званої функції ядра, що являється єдиним спірним моментом в даному методі, метод узагальнюється на випадок нелінійних роздільних поверхонь. Проте проблема вибору ядра водночас являється показником гнучкості методу: правильно підібране ядро дозволяє підлаштувати метод під найрізноманітніші задачі, фактично не змінюючи його суть.[21]

Машина опорних векторів (SVM) — це контрольований алгоритм бінарної класифікації машинного навчання. Маючи набір двох типів точок у N вимірах, SVM

генерує (N-1) мірну гіперплощину, щоб розділити ці точки на дві групи, як показано нижче:

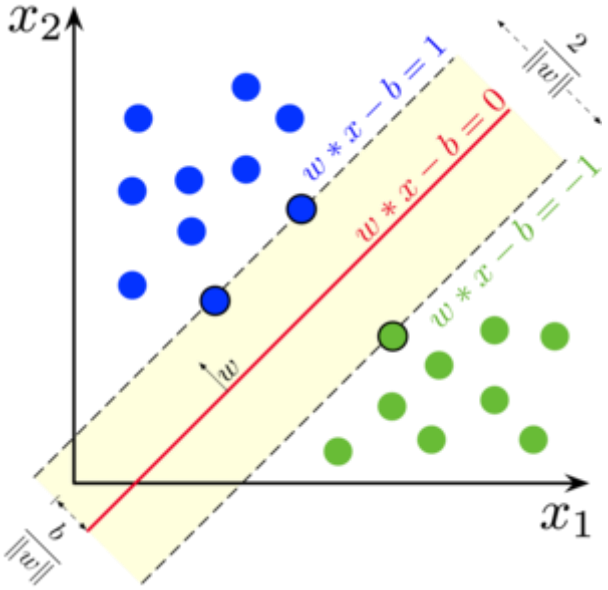


Рисунок 2.7. – графічне представлення алгоритму SVM

На наведеному вище малюнку SVM вибере червону лінію як найкращу гіперплощину, що розділяє синій і зелений класи.

Припустімо, у вас є два типи точок на площині, які лінійно розділяються. SVM знайде пряму лінію, яка розділяє ці точки на два типи та знаходиться якомога далі від усіх. Ця лінія відома як гіперплощина, і її вибрано так, щоб викиди не ігнорувалися, а точки різних класів були якомога далі одна від одної. Якщо точки неможливо розділити, SVM використовує перетворення ядра, щоб збільшити розміри точок.

SVM може виконувати не тільки лінійну класифікацію, а й нелінійну за допомогою «трюка з ядром» (англ. kernel trick), який показує неявно входи у багатовимірних просторах.

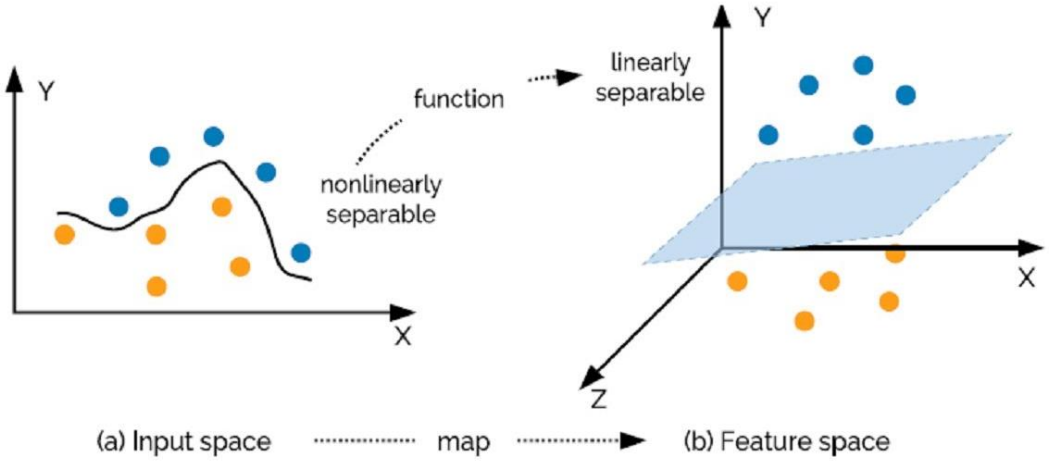


Рисунок 2.7. – графічне представлення “трюку з ядром”

Це створить тривимірний простір з попередніх пунктів. З малюнка нижче ми можемо зробити висновок, що спочатку точки не можна було розділити лінійно, але після застосування функції ядра ми легко розділили точки даних. Існує багато доступних функцій ядра, які ви можете вибрати відповідно до свого випадку використання.

Метод опорних векторів можемо сприймати як нелінійне узагальнення лінійного класифікатора, що заснований на розширенні розмірності простору передбачень за допомогою спеціальних ядерних функцій. За допомогою них можна будувати площини різноманітних форм.

Ядро SVM - це симетрична, додатно половинновизначена матриця K, що складається зі скалярних добутків пар:

$$x_i x_j : K(x_i x_j) = < f(x_i), f(x_j) >, \quad (2.8)$$

де f - довільна перетворююча функція для формування ядра. Наприклад:

1) лінійне ядро: $K(x_i x_j) = x_i^T x_j$

2) сигмоїдне ядро: $K(x_i x_j) = tan(\gamma\, x_i^T x_j \; + \beta 0)$

3) гаусове ядро з радіальною базовою функцією: $K(x_i x_j) = exp(\gamma ||x_i - x_j||^2)$

4) поліноміальне ядро зі степенем p: $K(x_i x_j) = (1 + x_i^T x_j)^p$

2.2.1. Недоліки методу опорних векторів і шляхи вдосконалення роботи методу.

Перелічимо основні недоліки методу Машини Опорних векторів і способи їх усунення і вдосконалення роботи методу.

Обчислювальна складність. Одним із істотних недоліків SVM є їх обчислювальна складність, особливо з великомасштабними наборами даних. Час навчання SVM може бути значно довшим порівняно з іншими алгоритмами машинного навчання, головним чином через проблему квадратичної оптимізації, яку вони вирішують. Зі збільшенням кількості навчальних зразків обчислювальне навантаження SVM стає більш вираженим. Це обмеження обмежує їх масштабованість і ефективність у сценаріях, де час є критичним фактором.

Стратегії вдосконалення:

Апроксимація ядра: одним із підходів до зменшення обчислювальної складності SVM є використання методів апроксимації ядра. Ці методи спрямовані на апроксимацію функції ядра та зменшення обчислювального навантаження без значної втрати продуктивності прогнозування. Приклади включають випадкові функції Фур'є та апроксимацію Ністрема, які забезпечують ефективні апроксимації ядерної матриці.

Розпаралелювання: використання методів паралельних обчислень може допомогти прискорити навчання SVM. Розподіляючи обчислювальне навантаження між кількома процесорами або використовуючи графічні процесори (GPU), час навчання SVM можна значно скоротити.

Чутливість до гіперпараметрів: SVM дуже чутливі до вибору гіперпараметрів, таких як параметр регулярізації (C) і параметри, специфічні для ядра. Вибір невідповідних значень для цих параметрів може призвести до неоптимальної продуктивності моделі або переобладнання.

Стратегії вдосконалення:

Пошук у сітці та перехресна перевірка: використання таких методів, як пошук у сітці та перехресна перевірка, може допомогти визначити оптимальні значення для гіперпараметрів. Пошук у сітці включає вичерпний пошук у попередньо визначеній сітці параметрів, оцінюючи продуктивність SVM за кожною комбінацією параметрів. Перехресна перевірка допомагає оцінити продуктивність узагальнення, розділяючи дані на кілька підмножин для навчання та перевірки, дозволяючи вибрати гіперпараметри, які дають найкращу середню продуктивність у різних складках.

Обробка класів, що перекриваються або є нероздільними: SVM можуть мати проблеми під час роботи з наборами даних, де класи перетинаються або нероздільні за допомогою лінійних меж рішень. У таких випадках SVM потребують більш складних функцій ядра або складних методів розробки функцій для точного захоплення базових шаблонів.

Стратегії вдосконалення:

Нелінійні ядра: використання нелінійних функцій ядра, таких як ядра поліноміальної або радіальної базисної функції (RBF), дозволяє SVM моделювати складні зв’язки між функціями та покращувати продуктивність нелінійно розділених наборів даних. Експериментування з різними функціями ядра може допомогти визначити найбільш підходящу для певної проблеми.

Розробка функцій: перетворення або розширення вхідних функцій може підвищити роздільність класів. Такі методи, як вибір функцій, зменшення розмірності (наприклад, аналіз основних компонентів) або включення знань предметної області можуть допомогти у створенні більш розрізнювальних функцій і покращенні продуктивності SVM.

Чутливість до викидів: SVM можуть бути чутливими до викидів, оскільки вони покладаються на опорні вектори поблизу межі прийняття рішення. Викиди, які потрапляють у межі поля або порушують межі поля, можуть впливати на продуктивність моделі. Приклад викиду можна побачити на рисунку 2.8.

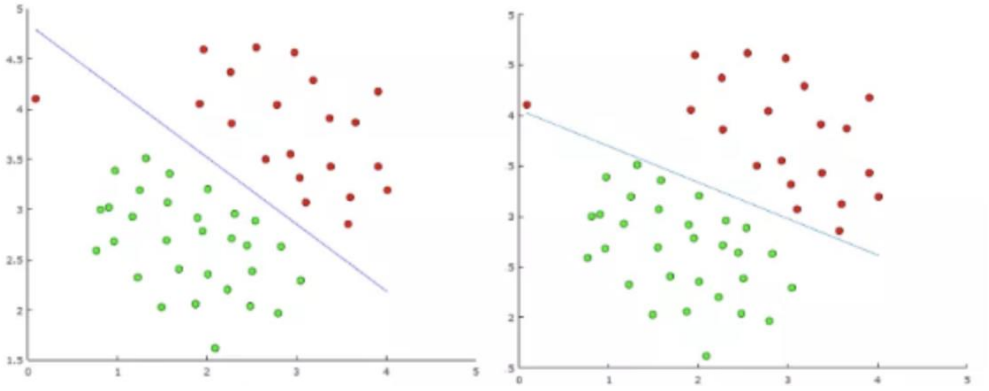


Рисунок 2.8. – графічне представлення “викиду” і перебудовування межі

Стратегії вдосконалення:

Виявлення викидів: до навчання SVM виявлення та обробка викидів за допомогою методів виявлення викидів може допомогти пом’якшити їхній вплив на модель. Викиди можна видалити з навчального набору даних або зменшити вагу під час процесу оптимізації.

Надійне масштабування функцій. Застосування методів надійного масштабування функцій, таких як медіанне або процентильне масштабування, може зробити SVM більш стійким до наявності викидів і підвищити його надійність.

Робота з великими наборами даних: SVM можуть зіткнутися з проблемами під час роботи з великомасштабними наборами даних, які не вміщуються в пам’ять або мають багаторозмірні простори функцій.

Стратегії вдосконалення:

Вибір підмножини: навчання SVM на підмножині даних може зменшити обчислювальне навантаження. Ретельні стратегії відбору підмножини, такі як випадкова вибірка або методи на основі кластеризації, можуть забезпечити репрезентативні навчальні вибірки.

Стохастичний градієнтний спуск: використання стохастичного градієнтного спуску замість вирішення повної проблеми оптимізації може бути корисним для

обробки великих наборів даних. Стохастичний градієнтний спуск випадковим чином відбирає підмножину навчальних прикладів у кожній ітерації, що призводить до швидшої конвергенції та зменшення вимог до пам’яті.

Згадаємо також інші вдосконалення роботи методу:

Методи ансамблю: використання методів ансамблю може покращити продуктивність і надійність SVM.

Пакетування: застосування методів пакетування, таких як випадкові ліси (Random Forests) або початкове агрегування (Bootstrap Aggregating), передбачає навчання кількох моделей SVM на різних підмножинах навчальних даних. Поєднання передбачень цих моделей може призвести до покращеного узагальнення та кращої обробки шумних або невизначених даних.

Прискорення: Алгоритми прискорення, такі як AdaBoost або Gradient Boosting, послідовно навчають слабкі моделі SVM, які зосереджуються на виправленні помилок попередніх моделей. Цей ітераційний процес може підвищити загальну продуктивність і зробити SVM більш адаптованим до складних наборів даних.

Включення знань предметної області: інтеграція знань предметної області в процес моделювання SVM може підвищити його продуктивність і вирішити конкретні проблеми, пов’язані з проблемою, що розглядається.

Вибір функцій: визначення пріоритетності відповідних функцій на основі знань домену може покращити продуктивність SVM шляхом зменшення шуму та зосередження на найбільш інформативних атрибутах.

Попередня обробка даних. Застосування методів попередньої обробки даних для конкретної області, таких як нормалізація, масштабування функцій або очищення даних, може покращити здатність SVM витягувати значущі шаблони та робити точні прогнози.

Підсумовуючи, машини опорних векторів (SVM) демонструють такі недоліки, як обчислювальна складність, чутливість до гіперпараметрів, труднощі в

обробці класів, що збігаються, і викиди, а також проблеми з великими наборами даних. Однак ці обмеження можна усунути за допомогою різних стратегій вдосконалення, включаючи апроксимацію ядра, розпаралелювання, налаштування гіперпараметрів, використання нелінійних ядер, розробку функцій, виявлення викидів, надійне масштабування функцій, вибір підмножини, стохастичний градієнтний спуск, ансамблеві методи та використання знань предметної області. Розглядаючи ці стратегії, дослідники та практики можуть пом’якшити обмеження SVM та покращити їхню продуктивність у різноманітних програмах.

РОЗДІЛ 3. ПОБУДОВА МОДЕЛІ АНАЛІЗУ І ПРОГНОЗУВАННЯ КОТУВАНЬ НА ФОНДОВІЙ БІРЖІ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

3.1. Формування бази даних котувань акцій різних компаній

Формалізація бази знань даних котувань акцій різних компаній є важливим кроком у покращенні аналізу та прогнозування цього типу даних. Цей процес включає в себе створення бази даних, яка містить інформацію про причини, симптоми, діагностику та лікування цього захворювання.

Формалізація бази даних дозволяє ефективніше використовувати отриману інформацію та забезпечити більш точний та швидкий аналіз даних. Крім того, вона сприяє покращенню зберігання та обробки даних, зниженню кількості помилок в аналізі та прогнозування та, як наслідок, покращенню інвестиційних стратегій та планів.

В ході роботи було вирішено скористатися даними, запропонованими на сайті Yahoo Finance[26], для аналізу тренування та тестування моделей і методів машинного навчання. Дані з вищеозначеного джерела мають наступні переваги:

Легко доступний: цей датасет доступний для безкоштовного завантаження та використання в дослідженнях.

Консолідована інформація: за допомогою ресурсу можна отримати доступ до історичних даних усіх компаній які представлені на Ню-Йоркській біржі.

Реалістична інформація: ресурс містить дані про змінення цін акцій на великому проміжку часу, що є реальними біржовими даними, а не вигаданими.

Структуровані дані: ресурс надає датасети, які легко буде у подальшому використовувати для аналізу і прогнозування без попередніх очищень і змін форматів.

Велика кількість змінних: ресурс містить велику кількість змінних, які можна використовувати для аналізу ризиків та іншої статистичної обробки даних.

Перевірений: дані з ресурсу використовувався в декількох наукових дослідженнях та вже пройшов перевірку на достовірність та точність.

Різноманітність даних: ресурс містить датасети про компанії з різним капіталом, обігом, що надають різні види послуг на ринку і вивели свої акції на біржу.

Yahoo Finance зберігає велику кількість історичних даних, пов'язаних з фінансовими ринками. Ці дані охоплюють різні типи фінансових інструментів, включаючи акції, індекси, товари, валюти та облігації. Історичні дані, доступні на Yahoo Finance, дозволяють користувачам аналізувати минулі ринкові тенденції, здійснювати технічний аналіз та розробляти торговельні стратегії.

Історичні дані, які надаються Yahoo Finance, зазвичай містять таку інформацію:

1. Дата: Дата торгової сесії, для якої записані дані.
2. Ціна відкриття: Ціна, за якою інструмент почав торгуватися на початку сесії.
3. Максимальна ціна: Найвища ціна, досягнута інструментом протягом торгової сесії.
4. Мінімальна ціна: Найнижча ціна, досягнута інструментом протягом торгової сесії.
5. Ціна закриття: Ціна, за якою інструмент закінчив торгуватися в кінці сесії.
6. Скоригована ціна закриття: Ціна закриття, скоригована на фактори, такі як дивіденди, поділ акцій або інші корпоративні дії.
7. Обсяг: Кількість акцій або контрактів, що були укладені протягом сесії.
8. Ринкова капіталізація: Ринкова капіталізація компанії, пов'язаної з інструментом.

Yahoo Finance надає історичні дані на різних проміжках часу, від щоденних до щотижневих, щомісячних або навіть внутріденних інтервалів. Наявність історичних даних на Yahoo Finance може варіюватися в залежності від інструменту та обраного проміжку часу.

Історичні дані на Yahoo Finance можна отримати у різних форматах, наприклад, у форматі CSV (Comma-Separated Values), які легко імпортувати до інструментів аналізу даних, таких як Excel або Python, для подальшого аналізу. Це дозволяє користувачам здійснювати статистичні розрахунки, генерувати діаграми та графіки та застосовувати різні кількісні методи для отримання висновків з історичних ринкових даних.

Крім того, Yahoo Finance пропонує додаткові функції та показники, отримані з історичних даних, такі як рухомі середні, технічні індикатори та фінансові показники. Ці інструменти можуть допомогти виявляти патерни, тенденції та потенційні торговельні можливості.

Важливо зазначити, що, хоча Yahoo Finance надає значну кількість історичних даних, наявність та повнота даних можуть варіюватися для різних фінансових інструментів та проміжків часу. Рекомендується завжди перевіряти дані з кількох джерел та забезпечувати їх точність та надійність перед прийняттям фінансових рішень на основі історичних даних.

Для аналізу були взяті реальні дані по котируванням акцій міжнародних високотехнологічних IT компаній:

- EPAM Systems, Inc. (EPAM). Кількість записів у таблиці 2836 (з 08/02/2012 із виключенням вихідних та святкових днів).
- Microsoft Corporation (MSFT). Кількість записів у таблиці 9370 (з 13/03/1986 із виключенням вихідних та святкових днів).
- Adobe Systems Incorporated (ADBE). Кількість записів у таблиці 9264 (з 13/08/1986 із виключенням вихідних та святкових днів).
- Cisco Systems, Inc. (CSCO). Кількість записів у таблиці 8375 (з 16/02/1990 із виключенням вихідних та святкових днів).
- Globant S.A. (GLOB). Кількість записів у таблиці 2223 (з 18/07/2014 із виключенням вихідних та святкових днів).

- Alphabet Inc. (GOOG). Кількість записів у таблиці 4718 (з 19/08/2004 із виключенням вихідних та святкових днів).
- International Business Machines Corporation (IBM). Кількість записів у таблиці 15449 (з 02/01/1962 із виключенням вихідних та святкових днів).
- Meta Platforms, Inc. (META). Кількість записів у таблиці 2766 (з 18/05/2012 із виключенням вихідних та святкових днів).
- Oracle Corporation (ORCL). Кількість записів у таблиці 9371 (з 12/03/1986 із виключенням вихідних та святкових днів).
- VMware, Inc. (VMW). Кількість записів у таблиці 3966 (з 17/08/2007 із виключенням вихідних та святкових днів).

За допомогою бібліотеки matplotlib, яку буде детальніше розглянуто у наступному розділі, я успішно побудував графік акцій десяти вищенаведених IT компаній у моїй дипломній роботі. Ця бібліотека відома своєю простотою використання та потужними можливостями для візуалізації даних.

Завдяки простому інтерфейсу matplotlib, я зміг швидко побудувати графік, використовуючи доступні функції та методи. Ця бібліотека надає широкий спектр інструментів для керування виглядом графіків, включаючи налаштування осей, легенди, заголовка та інших елементів. Загальна простота використання бібліотеки matplotlib допомогла мені ефективно візуалізувати дані про акції IT компаній у моїй дипломній роботі. Це надало мені можливість проаналізувати залежності, тенденції та інші важливі аспекти цих акцій.

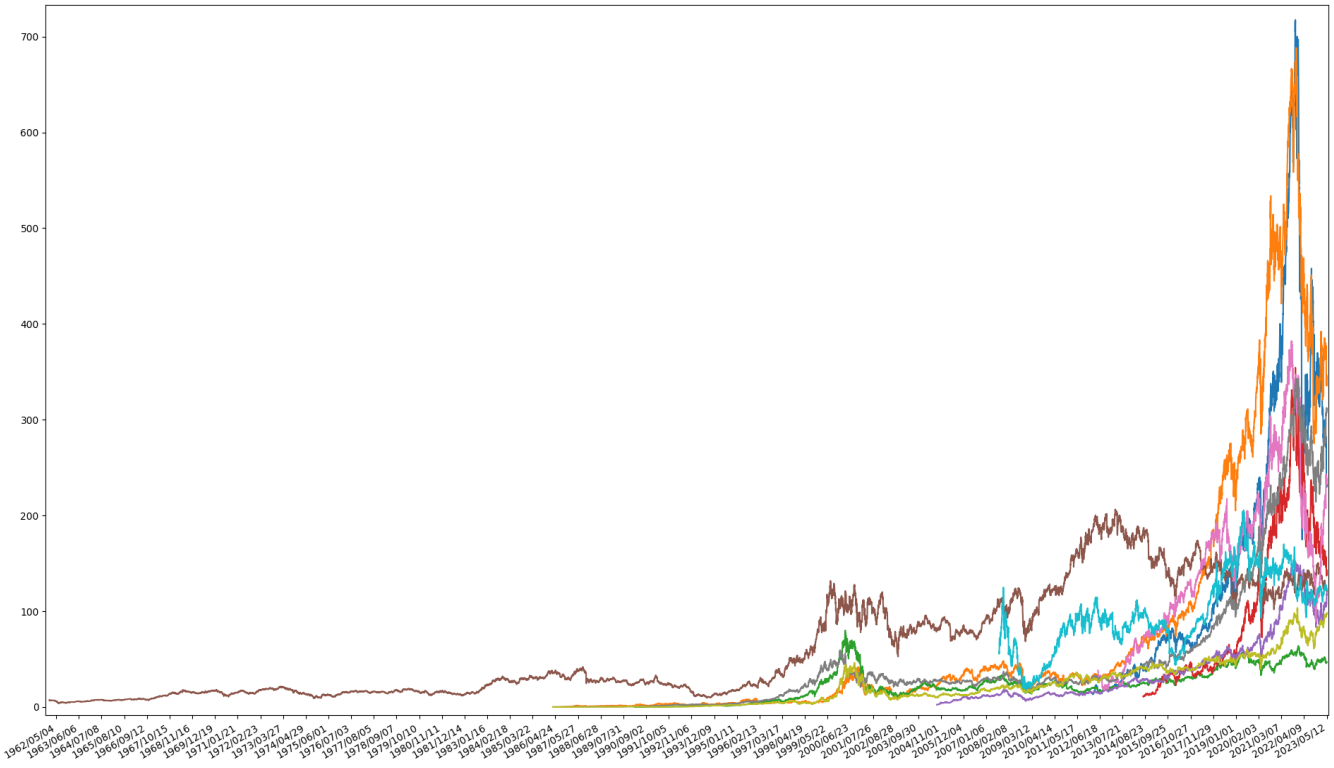


Рисунок 3.1 – Графік котувань акцій всіх компаній

На рисунку 3.1 бачимо графік котирувань акцій кожної з 10 компаній які були відібрані для подальшого дослідження. Можна звернути увагу на загальні тренди активного зростання IT індустрії з початку 1997 року. А потім спад на початку 2001 року. Світова фінансова криза 2008 року на мала істотного впливу на IT індустрію, хоч і можна помітити невелике зниження курсу всіх акцій. З 2014 року можна побачити швидке зростання цін на акцій майже всіх представлених компаній. І швидке падіння наприкінці 2021 – початку 2022 років.

На загальному графіку представлені компанії кожна з яких позначена своїм кольором. Найдовшою кривою є коричнева, що відповідає компанії International Business Machines Corporation (IBM). Наступним є світло-зелена крива який відображає коливання акцій компанії Oracle Corporation (ORCL), які були представлені на біржі 12 Березня 1986. На наступний день, 13 Березня 1986 року на Ню-Йоркську біржу були представлені акції компанії Microsoft Corporation (MSFT), які відображені на графіку сірою кривою. Рівно за 5 місяців були представлені акції компанії Adobe Systems Incorporated (ADBE).

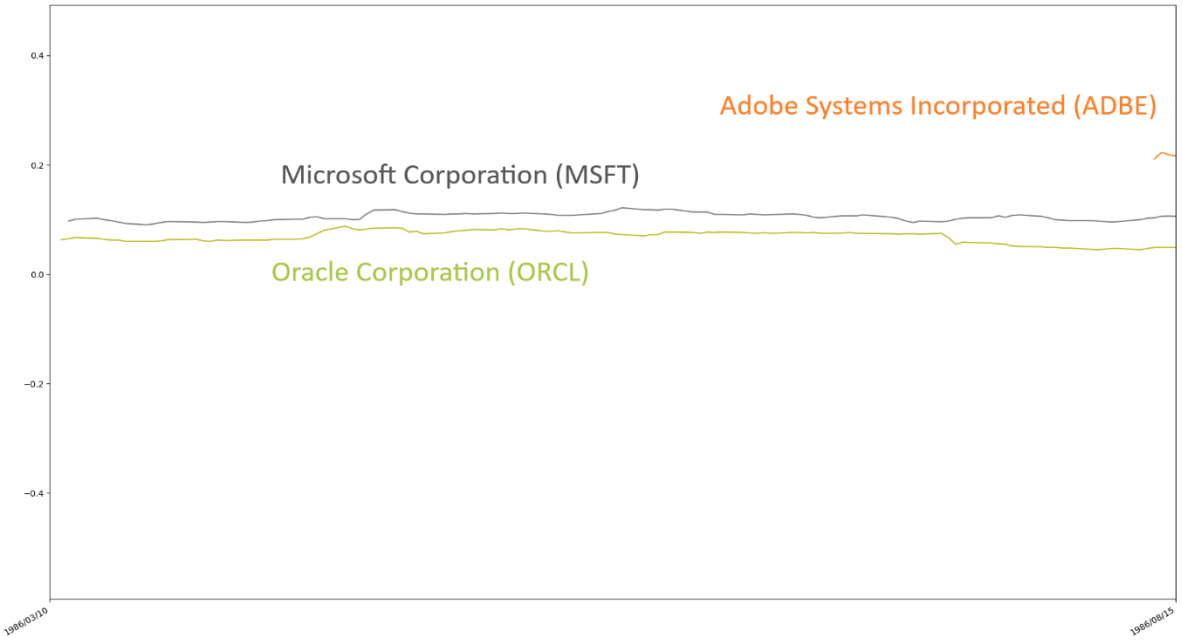


Рисунок 3.2 – Графік котувань акцій компаній ORCL, MSFT, ADBE

Далі компанії йдуть у наступній часово-кольоровій послідовності: Cisco Systems, Inc. (CSCO) – темно-зелений колір, Alphabet Inc. (GOOG) – фіолетовий колір, VMware, Inc. (VMW) – бірюзовий, EPAM Systems, Inc. (EPAM) – синій колір, Meta Platforms, Inc. (META) – рожевий колір, Globant S.A. (GLOB) – червоний колір.

Загалом подібне відображення має скоріше функцію візуалізації вже наявної інформації, а не отримання нових знань та інсайтів. Тим не менш на мою думку було важливо відобразити у роботі цю частину, бо візуалізовані дані є більш простими для сприйняття і допомагають скласти загальну картину.

3.2. Вибір засобів для реалізації моделі

Для реалізації безпосередньо задачі аналізу і прогнозування котирувань акцій було обрано стандартні інструменти, які використовуються в таких сферах, як Машинне навчання, Data Science, Business Intelligence та суміжних до них. Далі буде детально розглянуто кожний з цих інструментів і технологій.

3.2.1. Мова програмування Python

У якості мови програмування для реалізації проекту був обраний “python”. Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. [24] Розроблена в 1990 році Гвідо ван Россумом. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна та функціональна.

Серед основних її переваг можна назвати такі:

- Чистий та інтуїтивно зрозумілий синтаксис;
- Переносність програм (що властиве більшості інтерпретованих мов);
- Стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- Можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- Зручний для розв'язання математичних проблем;
- Доступність великої кількості бібліотек пов’язаних з ШІ та ШНМ.

Python є високорівневою мовою програмування, що, заради універсальності в підході до програмування, використовує різні парадигми програмування, зокрема:

- об’єктно-орієнтованість – парадигма програмування на основі множини об’єктів, що взаємодіють на основі концепцій інкапсуляції, спадкування, поліморфізму та абстракції;
- процедурність – парадигма реалізації послідовних кроків заснованих на підпрограмах, методах або функціях, що викликаються з будь-якого місця програми, включно із самовикликом такого коду через рекурсію;

- функціональність – парадигма, що заснована виключно на обробці функцій та уникає стану зміни даних, а способом розбиття програми є створення нової функції при правилі композиції – оператора суперпозиції функції;
- аспектно-орієнтованість – парадигма доповнення або альтернатива ООП, що вводить ділення функціональності на класи або модулі та впроваджує додаткову логіку функціональності, що викликається у місцях з'єднання таких модулів.

Це інтерпретована мова високого рівня, що використовує принципи строгої динамічної типізації, що дозволяє використовувати гнучкість багатьох скриптових мов програмування – роботу із змінними, що не мають типів. Такий підхід дозволяє використовувати змінні переписуючи в них дані не одного типу в залежності від необхідності логіки роботи програми.

Перевагами використання мови Python є швидкість написання програмного забезпечення, чому сприяє широка підтримка модулів, бібліотек та готових програмних пакетів над якими працюють розробники, що входять до спільноти підтримки цієї мови програмування.

3.2.2. Репозиторій GitHub

Для зберігання, контролю змін, відстеження проблем було обрано репозиторій GitHub. GitHub — це веб-платформа, створена на основі Git, розподіленої системи контролю версій, яка широко використовується в розробці програмного забезпечення. Він пропонує низку функцій і можливостей, які полегшують спільне кодування, керування проектами та обмін кодом. GitHub здобув величезну популярність серед розробників та організацій завдяки простоті використання, надійній інфраструктурі та активній спільноті.

Основні характеристики GitHub:

- Хостинг репозиторіїв: GitHub дозволяє розробникам створювати та розміщувати репозиторії Git для своїх проектів.

- Контроль версій: Git, базова технологія GitHub, дозволяє розробникам відстежувати та керувати змінами у своїй базі коду.
- Інструменти для співпраці: GitHub пропонує кілька функцій для співпраці, як-от запити на отримання, перегляд коду та відстеження проблем.
- Відстеження проблем: GitHub надає систему відстеження проблем, яка дозволяє розробникам повідомляти про помилки, пропонувати функції або створювати завдання.
- Соціальні функції: GitHub має сильний соціальний аспект, що дозволяє розробникам стежити за проектами.
- Безперервна інтеграція/розгортання (CI/CD): GitHub інтегрується з різними інструментами CI/CD.
- Документація та Wiki: GitHub містить функції для документування проектів через wiki та сторінки GitHub.

У підсумку можна сказати, що GitHub революціонізує спосіб співпраці розробників, керування кодом і внесок у проекти програмного забезпечення. Його потужна система контролю версій, інструменти для співпраці та велика спільнота роблять його популярною платформою для розробників у всьому світі. GitHub надає інфраструктуру та функції, необхідні для успішної та ефективної розробки програмного забезпечення. Репозиторій був дуже корисним для імплементації розробленого рішення.

3.2.3. Інтегроване середовище розробки VSCode

У сучасному програмуванні інтегровані середовища розробки (Integrated Development Environments - IDEs) є невід'ємною частиною процесу розробки програмного забезпечення. Вони надають розробникам зручні інструменти для написання, редагування, відлагодження та тестування коду. У цьому розділі я детально розгляну використання одного з популярних інтегрованих середовищ розробки - Visual Studio Code (VSCode), яке допомогло мені успішно написати

програму для моєї дипломної роботи. Переваги використання VSCode у розробці програмного забезпечення:

- Налаштовуваність та розширюваність: Можливість налаштувати середовище розробки під свої потреби за допомогою розширень та налаштувань.
- Підтримка різних мов програмування: VSCode підтримує широкий спектр мов програмування та надає інструменти для роботи з ними.
- Інтеграція з системами керування версіями: Можливість безпосередньо працювати з Git, SVN та іншими системами керування версіями.

VSCode дозволив мені зручно та ефективно розробляти програму, надавши широкі можливості для редагування, відлагодження та тестування коду. Переваги VSCode включають його зручний інтерфейс, підтримку різноманітних мов програмування, зокрема python, багатофункціональність та можливість інтеграції з іншими інструментами розробки.

Завдяки зручному інтерфейсу та розширеним можливостям редагування коду я міг швидко та ефективно розробляти програму. Вбудовані інструменти для відлагодження дозволили мені виявити та виправити помилки у програмі, забезпечивши її більшу стабільність та надійність. Крім того, засоби для тестування допомогли перевірити правильність реалізації функцій та забезпечити якість програми.

Загалом, використання інтегрованого середовища розробки VSCode виявилося надзвичайно корисним у процесі розробки програми для моєї дипломної роботи. Він забезпечив зручність, продуктивність та можливості для спільної роботи, що дозволило мені успішно впоратися зі складностями розробки та створити високоякісну програму.

3.2.4. Бібліотека програмного забезпечення TensorFlow

TensorFlow — це бібліотека програмного забезпечення з відкритим вихідним кодом для високопродуктивних чисельних обчислень. Його гнучка архітектура

дозволяє легко розгортати обчислення на різних платформах (CPU, GPU, TPU) і від настільних комп’ютерів до кластерів серверів до мобільних і граничних пристроїв.[25]

Бібліотека програмного забезпечення TensorFlow виявилася незамінною у моїй дипломній роботі, особливо при роботі з нейронними мережами. Вона забезпечила мені потужні інструменти для розробки, навчання та використання нейронних мереж.

Завдяки TensorFlow я зміг легко створити архітектуру нейронної мережі, використовуючи його високорівневі API. Бібліотека надала мені доступ до різних типів шарів, функцій активації та оптимізаторів, що дозволило налаштувати модель під мої потреби.

Один із найбільших переваг TensorFlow - це його потужний обчислювальний граф. Він дозволив мені ефективно працювати з великими обсягами даних та виконувати розподілені обчислення на різних пристроях. Це було особливо корисно при навчанні нейронної мережі на графічних процесорах (GPU), що значно прискорило процес.

Усі ці можливості TensorFlow відіграли ключову роль у моїй дипломній роботі, допомагаючи мені створити, навчити та оцінити ефективні нейронні мережі. Використання цієї бібліотеки спростило процес розробки, забезпечивши широкий набір інструментів та ресурсів для досягнення моїх цілей.

3.2.5. Бібліотека нейронних мереж Keras

Keras — відкрита нейромережна бібліотека, написана мовою Python. Вона здатна працювати поверх TensorFlow, Microsoft Cognitive Toolkit, R. Спроєктовану для уможливлення швидких експериментів з мережами глибинного навчання, її зосереджено на тому, щоби вона була зручною в користуванні, модульною та розширюваною.

Одна з головних переваг Keras полягає в його простоті та легкості використання. Бібліотека надає простий та інтуїтивно зрозумілий інтерфейс для створення нейронних мереж без необхідності в глибоких знаннях про математичні моделі та алгоритми.

Крім того, Keras забезпечує високий рівень абстракції, що дозволяє швидко та ефективно розробляти складні архітектури нейронних мереж. За допомогою Keras, я міг швидко створити багатошарові мережі, налаштувати функції активації та оптимізатори, а також використовувати зручний API для навчання та оцінки мережі.

Ще одна важлива особливість Keras - це його можливість інтеграції з різними обчислювальними бекендами, зокрема TensorFlow, який я вже згадував. Це дозволяє використовувати всю потужність цих бібліотек для обчислень нейронних мереж.

Загалом, використання бібліотеки Keras допомогло мені у дипломній роботі, дозволяючи швидко та ефективно розробляти та налаштовувати нейронні мережі. Його простота використання та гнучкість роботи зробили процес розробки мереж більш доступним та продуктивним.

3.2.6. Програмна бібліотека машинного навчання Scikit-learn

Scikit-learn (також відома як sklearn або scikits.learn) — це безкоштовна програмна бібліотека машинного навчання для мови програмування Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest, градієнтний бустинг та інші. Scikit-learn виявилася незамінною у моїй дипломній роботі, особливо при використанні методу опорних векторів.[29]

Scikit-learn надав мені потужні інструменти для реалізації та застосування методу опорних векторів (SVM). Цей метод використовується для задач

класифікації та регресії і заснований на пошуку гіперплощини, яка найкраще розділяє дані різних класів.

За допомогою Scikit-learn я міг легко створити модель SVM, налаштувати гіперпараметри та виконати навчання моделі на моїх даних. Бібліотека надала мені доступ до різних типів ядер (наприклад, лінійне, поліноміальне, радіальне базисне функції), які дозволяють моделі адаптуватися до різних типів даних та складних структур.

Крім того, Scikit-learn має широкий набір утиліт для попередньої обробки даних, таких як масштабування, вибір ознак та перехресна перевірка. Ці інструменти допомогли мені покращити якість моделі SVM шляхом оптимізації параметрів та обробки даних перед навчанням.

Загалом, використання бібліотеки Scikit-learn допомогло мені в дипломній роботі з методом опорних векторів. Його простота використання, багатий функціонал та підтримка різних методів машинного навчання роблять Scikit-learn незамінним інструментом для розв'язання завдань машинного навчання.

3.2.7. Програмна бібліотека Pandas

Pandas — програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами.

Pandas надавала мені зручні методи для завантаження, обробки та аналізу даних у форматі датафреймів. Я міг легко імпортувати дані з різних джерел, таких як CSV-файли та xls-файли, та виконувати операції з ними, такі як фільтрація, сортування, групування та обчислення агрегованих статистик.

Одна з найважливіших властивостей Pandas - це можливість легко інтегруватися з іншими бібліотеками, такими як Scikit-learn і Keras. Pandas дозволяє перетворювати датафрейми у формати, які необхідні для цих бібліотек, та зручно передавати дані між ними. Наприклад, я міг використовувати Pandas для підготовки

та очищення даних, а потім передавати їх у моделі Scikit-learn для навчання або у моделі Keras для розробки нейронних мереж.

У підсумку, бібліотека Pandas була важливим компонентом моєї дипломної роботи, допомагаючи мені ефективно працювати з датафреймами даних, взаємодіяти з бібліотеками Scikit-learn і Keras та забезпечувати потрібну підготовку даних для моделей машинного навчання. Вона надала мені широкі можливості для завантаження, обробки, аналізу та маніпулювання даними, що були важливі для моєї дипломної роботи.

3.2.8. Програмна бібліотека matplotlib

matplotlib — бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Бібліотека відіграла важливу роль у моїй дипломній роботі на тему "Прогнозування котувань на фондовій біржі методами машинного навчання". matplotlib є потужним інструментом для візуалізації даних і побудови графіків.

Під час моїх досліджень, matplotlib дозволила мені створити високоякісні графіки, діаграми та інші візуальні представлення, що допомогли аналізувати і відображати залежності та тренди в котуваннях на фондовій біржі. Завдяки різноманітним функціям бібліотеки, я зміг створити лінійні графіки, гістограми, розсіювальні діаграми та інші типи візуалізацій, які показалися надзвичайно корисними для досліджень моєї теми.

Крім того, matplotlib була використана для візуалізації результатів моїх прогнозів та порівняння їх з фактичними котируваннями на фондовій біржі. Це дозволило мені оцінити ефективність моїх моделей прогнозування та зробити висновки щодо їхньої точності та надійності.

Загалом, завдяки багатофункціональності та зручному інтерфейсу matplotlib, я зміг зробити візуалізацію моїх даних та результатів прогнозування більш

інформативною та зрозумілою. Це сприяло кращому розумінню залежностей у котуваннях на фондовій біржі та підкреслило результати моїх досліджень.

3.3. Підготовка середовища та серверної частини до програмування штучної нейронної мережі та методу опорних векторів

Перш за все був налаштований репозиторій, до якого я доєднав локальні файли і папки. Використання репозиторію GIT стало потужним інструментом для розробки та безпечного збереження коду, спільної роботи з командою та зручного керування версіями. Я зміг легко створити та керувати гілками розвитку, вносити зміни, відстежувати їх історію, вирішувати конфлікти і злиття, а також досліджувати попередні версії коду.

Далі було налаштовано сервер. Попередньо була створена віртуальна машина. На неї була встановлена операційна система Windows 10 Server. Далі були встановлені усі інструменти які були необхідні для компіляції коду, зокрема: VSCode, python, бібліотеки використані для реалізації проекту. Також були завантажені файли з датасетами, для подальшої обробки.

3.4. Моделі машинного навчання для прогнозування котувань акцій

Для реалізації було обрано нейронні мережі та метод опорних векторів. Для кожної з моделей буде продемонстрована реалізація, навчання, і тестування. А в кінці буде проведено порівняння успішності моделей.

3.4.1 Ініціалізація моделі нейронних мереж

Для побудови моделі скористаємося бібліотекою Keras. Будемо будувати багатошарову перцептронну архітектуру на основі регресії та із використанням алгоритму зворотного поширення похибки.

Перш ніж розпочати, давайте спочатку імпортуємо всі функції та класи, які ми збираємося використовувати. Це передбачає робоче середовище VSCode з встановленою бібліотекою глибокого навчання Keras.

```
from keras.models import Sequential
from keras.layers import Dense
```

Для полегшення завдання будемо формулювати проблему прогнозування часових рядів як проблему регресії. Тобто, враховуючи котування (ціну акції в доларах США) цього дня, будемо вираховувати, яка ціна буде наступного дня.

Ми можемо написати просту функцію для перетворення нашого одного стовпця даних у набір даних із двох стовпців. Перший стовпець містить кількість ціну в день (t), а другий стовпець містить ціну в наступний день (t+1), яку слід передбачити.

Функція приймає два аргументи: набір даних, який є масивом NumPy, який ми хочемо перетворити в набір даних, і *look_back*, який є кількістю попередніх кроків часу, які можна використовувати як вхідні змінні для прогнозування наступного періоду часу, у цьому випадку за замовчуванням за замовчуванням 1.

За замовчуванням буде створено набір даних, де X — котування акцій у певний момент часу (t), а Y — котування акцій у наступний раз (t + 1). Його можна налаштувати, змінивши значення *look_back.*

```
# convert an array of values into a dataset matrix
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+look_back), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return np.array(dataX), np.array(dataY)
```

Після того як ми змоделюємо наші дані та оцінимо навички нашої моделі на наборі навчальних даних, нам потрібно отримати уявлення про навички моделі на нових невидимих даних. Для звичайної задачі класифікації або регресії ми б зробили це за допомогою перехресної перевірки.

Для даних часових рядів важлива послідовність значень. Простий метод, який ми можемо використати, — це розділити впорядкований набір даних на набори даних для навчання та тестування. Наведений нижче код обчислює індекс точки

розщеплення та розділяє дані на навчальні набори даних із 80% спостережень, які ми можемо використовувати для навчання нашої моделі, залишаючи решту 20% для тестування моделі.

```
27  # split into train and test sets
28  train_size = int(len(dataset) * 0.8)
29  test_size = len(dataset) - train_size
30  train = dataset[0:train_size,:]
31  test = dataset[train_size:len(dataset),:]
```

Далі можемо використати данні для тренування і дані для тестування для розділення їх за допомогою функції *"create_dataset()"*. Під час моделювання було обрано look_back = 3.

```
32  # reshape dataset
33  look_back = 3
34  trainX, trainY = create_dataset(train, look_back)
35  testX, testY = create_dataset(test, look_back)
```

Тепер ми можемо побудувати багатошарову модель перцептрона для навчання даних. Ми використовуємо просту мережу з 1 вхідним, 1 прихованим шаром з 12 нейронами і вихідним шаром. Модель підходить за допомогою середньоквадратичної помилки, яка, якщо взяти квадратний корінь, дає нам оцінку помилки в одиницях набору даних. Були опробуванні кілька грубих параметрів і зупинився на наведеній нижче конфігурації, але в жодному разі вказана мережа не оптимізована.

```
36  # create and fit Multilayer Perceptron model
37  model = Sequential()
38  model.add(Dense(12, input_dim=look_back, activation='relu'))
39  model.add(Dense(1))
40  model.compile(loss='mean_squared_error', optimizer='adam')
41  model.fit(trainX, trainY, epochs=100, batch_size=2, verbose=2)
```

У 38 строчці коду бачимо використання активаційної функції "relu". У нейронній мережі функція активації відповідає за перетворення підсумованого зваженого входу від вузла в активацію вузла або вихід для цього входу.

Функція випрямленої лінійної активації (rectified linear activation function) або скорочено ReLU — це кусково-лінійна функція, яка виводить вхідний сигнал

безпосередньо, якщо він позитивний, інакше він виводить нуль. Вона стала функцією активації за замовчуванням для багатьох типів нейронних мереж, оскільки модель, яка використовує її, легше тренується і часто досягає кращої продуктивності.[27]

У 40 строчці коду бачимо використання у якості функції втрат – середньо-квадратичну помилку. Середня квадратична помилка, або MSE, є функцією втрат, яка використовується для задач регресії за замовчуванням.

Математично, це є бажаною функцією втрат за основою висновку з максимальною ймовірністю, якщо розподіл цільової змінної є гаусівським. Це функція втрат, яку потрібно оцінити першою і змінити лише за наявності поважної причини. Середня квадратична похибка розраховується як середнє з квадратів різниць між прогнозованим і фактичним значеннями. Результат завжди позитивний, незалежно від знака прогнозованого та фактичного значень, а ідеальне значення дорівнює 0,0. Зведення в квадрат означає, що більші помилки призводять до більшої кількості помилок, ніж менші, а це означає, що модель карається за більші помилки.

Функцію втрати середнього квадрата помилки можна використовувати в Keras, вказавши «mse» або «mean_squared_error» як функцію втрат під час компіляції моделі.[28]

Adam — це алгоритм оптимізації, який можна використовувати замість класичної процедури стохастичного градієнтного спуску для ітераційного оновлення ваг мережі на основі навчальних даних.

Алгоритм оптимізації Adam є розширенням стохастичного градієнтного спуску, який нещодавно отримав широке застосування для програм глибокого навчання в комп’ютерному баченні та обробці природної мови.

Adam відрізняється від класичного стохастичного градієнтного спуску. Замість того, щоб адаптувати швидкість навчання параметрів на основі середнього першого моменту (середнього), як у розповсюдженні середнього квадратичного

значення, Adam також використовує середнє значення других моментів градієнтів (нецентрована дисперсія).[28]

batch_size – кількість зразків на оновлення градієнта. Якщо не вказано, batch_size за замовчуванням буде 32.

epochs – кількість епох для навчання моделі. Епоха — це ітерація над усіма наданими даними x і y. Модель не навчається для певної кількості ітерацій, заданих епохами, а лише до досягнення епохи індексних епох.

verbose – це режим детальності. 0 = тихий, 1 = індикатор прогресу, 2 = один рядок на епоху. "auto" за замовчуванням має значення 1 у більшості випадків, а 2 при використанні з ParameterServerStrategy. Зауважте, що індикатор перебігу не є особливо корисним під час реєстрації у файлі, тому verbose=2 рекомендується, якщо він не працює в інтерактивному режимі (наприклад, у виробничому середовищі).[28]

3.4.2. Отримання прогнозів після використання нейронних мереж

Нарешті, ми можемо генерувати прогнози, використовуючи модель як для набору даних на яких проводилося тренування моделі, так і для тестових даних, щоб отримати візуальну індикацію навичок моделі. Детальніше див. Додаток Б.

```
58  # plot baseline and predictions
59  plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%d/%Y'))
60  plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=98))
61  plt.plot(x_dates, dataset)
62  plt.plot(x_dates, trainPredictPlot)
63  plt.plot(x_dates, testPredictPlot)
64  plt.gcf().autofmt_xdate()
65  plt.show()
```

Через те, як був підготовлений набір даних, ми повинні змістити передбачення так, щоб вони вирівнювалися по осі *x* з вихідним набором даних. Після підготовки дані наносять на графік, показуючи вхідний набір даних синім кольором, прогнози для набору даних на яких проводилося тренування – помаранчевим, прогнози для

невидимого тестового набору зеленим. Надалі будуть приведені лише дані для компанії EPAM, як для бази практики. Графіки інших компаній можна подивитись у додатку В.

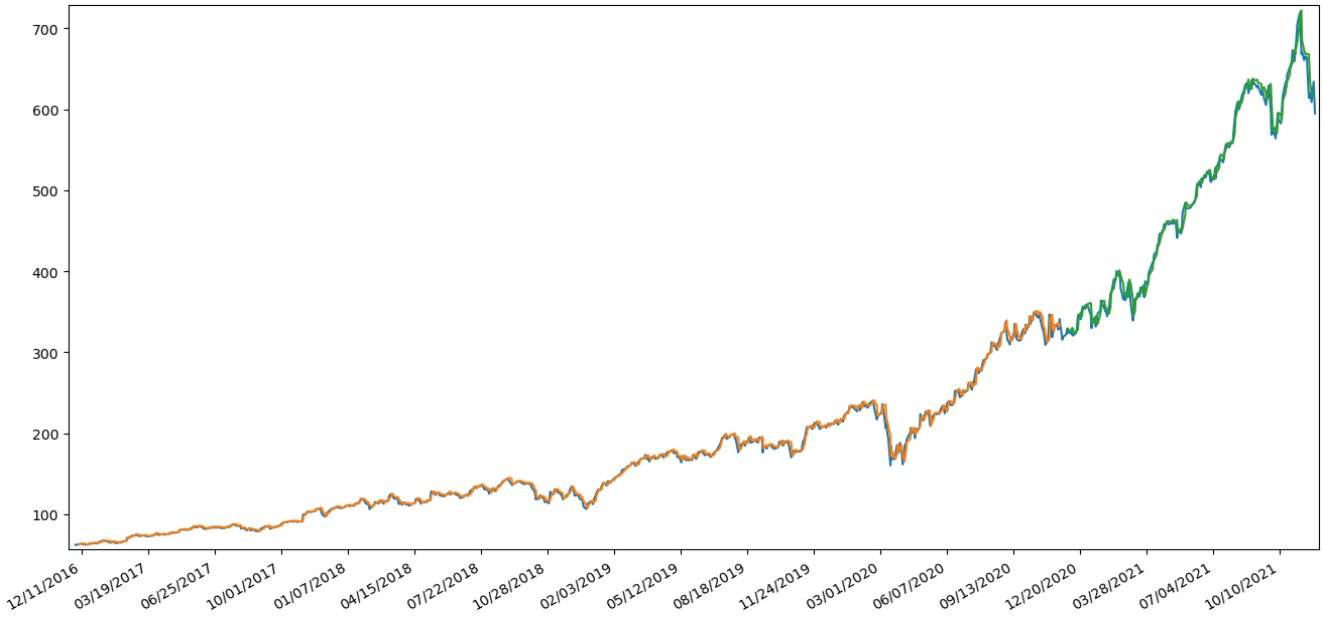


Рисунок 3.3 – Результати дослідження для компанії EPAM

3.4.3. Оцінка якості моделі нейронних мереж

Взявши квадратний корінь з оцінок продуктивності, ми бачимо, що модель має середню помилку 4.51 долари США у наборі даних для навчання та 10.96 доларів США у тестовому наборі даних. Це є 1,5% від локального максимуму і 3,5% від локального мінімуму. Точність моделі 96,5%-98,5%. В середньому 97,5%. Детальніше див. Додаток Б.

```
Train Score: 20.37 MSE (4.51 RMSE)
Test Score: 120.23 MSE (10.96 RMSE)
```

Отримані дані можна оцінити як успішну роботу нейронної мережі. Коливання середньої помилки можна пов’язати із зростанням ціни на акції та збільшенням амплітуди та розмаху коливань. Проте на рисунку 3.4 можна побачити, що прогнозування залишається доволі точним. Досягти цього вдалося в першу чергу шляхом спрощення задачі від прогнозування часових рядів до проблеми регресії.

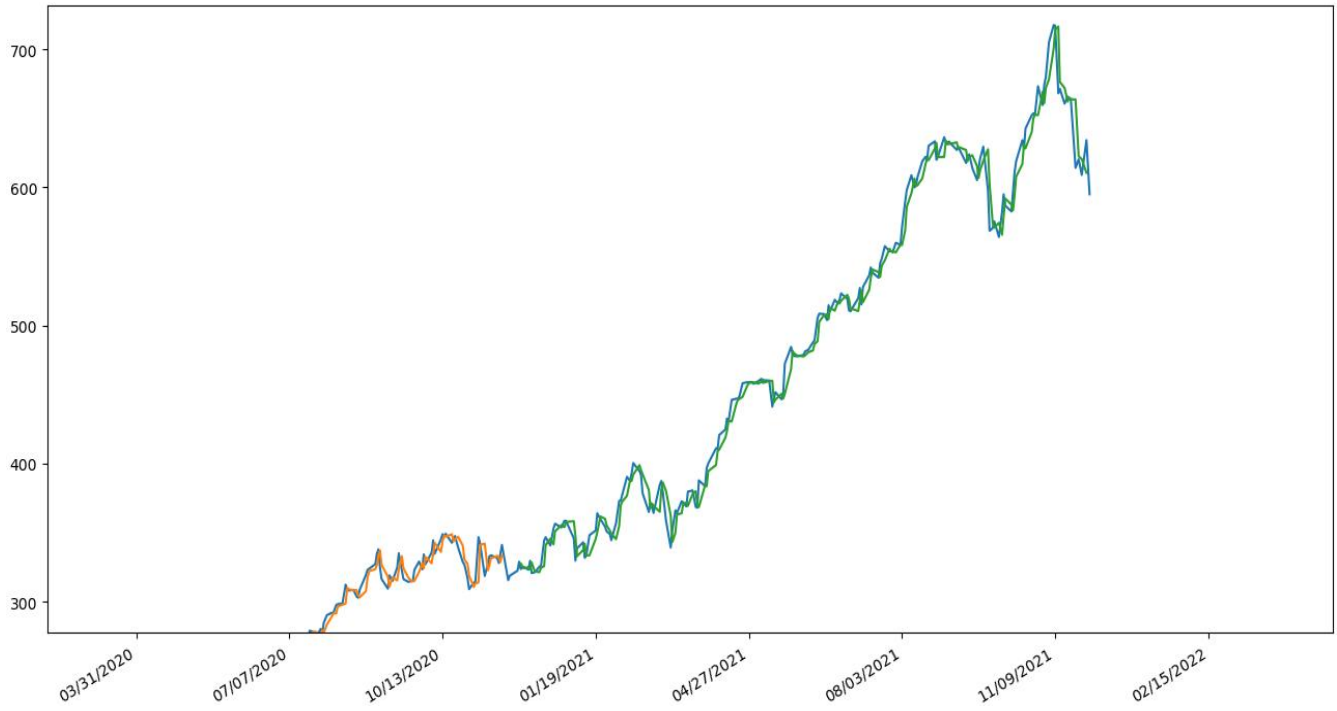


Рисунок 3.4 – Детальний розгляд прогнозу для компанії EPAM

Безумовно таке представлення є не більше ніж математичною моделлю і в реальній прикладній задачі націленій на прогнозування даних може не виправдати очікувань. Для отримання більш точних передбачень необхідно більше детально досліджувати ринок цінних паперів. Також необхідно проводити аналіз не лише за однією метрикою, але досліджувати також і додаткові дані присутні в таблиці, яка слугувала джерелом даних. Також необхідно приймати до уваги зовнішні чинники на ринку і поза ринком (так званих чорних лебедів).

3.4.4. Ініціалізація методу опорних векторів

Для побудови моделі скористаємося бібліотекою Sklearn. Будемо будувати модель методом опорних векторів.

Перш ніж розпочати, давайте спочатку імпортуємо всі функції та класи, які ми збираємося використовувати. Це передбачає робоче середовище VSCode з встановленою бібліотекою глибокого навчання Sklearn.

```
1
2   # Machine learning
3   from sklearn.svm import SVC
4   from sklearn.metrics import mean_squared_error
5
```

Для подальшого аналізу дані були піддані обробці з метою використання стовпця дати як індексу.

```
34  df_EPAM.index = pd.to_datetime(df_EPAM['Date'])
35
36  df = df_EPAM.drop(['Date'], axis='columns')
37
38  df['Open-Close'] = df.Open - df.Close
39  df['High-Low'] = df.High - df.Low
40
41  X = df[['Open-Close', 'High-Low']]
```

Пояснювальні або незалежні змінні використовуються для прогнозування значення змінної відповіді. Набір даних X складається з різних змінних, таких як "Відкриття-Закриття" та "Високе-Низьке". Ці змінні можна розглядати як індикатори, на основі яких алгоритм здійснює передбачення майбутнього тренду. Додавання додаткових індикаторів та оцінка їх ефективності також варто розглянути.

```
43  # Target variables
44  y = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)
45  split_percentage = 0.8
46  split = int(split_percentage*len(df))
47
48  # Train data set
49  X_train = X[:split]
50  y_train = y[:split]
51
52  # Test data set
53  X_test = X[split:]
54  y_test = y[split:]
```

Цільова змінна є результатом, який модель машинного навчання буде прогнозувати на основі пояснювальних змінних. y є набором даних цільової

змінної, в якому зберігаються правильні торгові сигнали, які алгоритм машинного навчання буде намагатися передбачити. Якщо ціна на завтра перевищує ціну на сьогодні, то ми купуємо певний акціонерний пакет, в іншому випадку ми не маємо жодної позиції. У змінній y ми зберігатимемо значення +1 для сигналу покупки і 0 для відсутності позиції. Для цього ми скористаємося функцією where() з бібліотеки NumPy. Ми також розділимо дані на навчальні та тестові набори даних. Це робиться для того, щоб ми могли оцінити ефективність моделі в тестовому наборі даних.

Ми будемо використовувати функцію SVC() з бібліотеки sklearn.svm.SVC, щоб створити нашу модель класифікатора за допомогою методу fit() для навчального набору даних.

```
56  # Support vector classifier
57  cls = SVC().fit(X_train, y_train)
```

sklearn.svm.SVC є класом, який надає реалізацію методу опорних векторів (Support Vector Machines, SVM) для задач класифікації. SVM є потужним алгоритмом машинного навчання, який може використовуватися для розв'язання задач класифікації, які включають як бінарні, так і багатокласові сценарії.

Після ініціалізації об'єкта SVC можна викликати методи для навчання моделі та здійснення передбачень на нових даних. Для навчання моделі на доступних даних можна використовувати метод fit(X_train, y_train), де X_train представляє собою набір даних з незалежними змінними, а y_train - відповідні мітки класів або значення цільової змінної. Метод fit виконує оптимізацію моделі шляхом знаходження оптимальних параметрів, які дозволяють розділити класи вхідних даних з найбільшою точністю.

```
59  df['Predicted_Signal'] = cls.predict(X)
60  # Calculate daily returns
61  df['Return'] = df.Close.pct_change()
62  # Calculate strategy returns
63  df['Strategy_Return'] = df.Return *df.Predicted_Signal.shift(1)
```

Після навчання моделі можна використовувати метод predict(X) для здійснення передбачень на нових даних X. Цей метод повертає прогнозовані мітки класів або значення цільової змінної для нових даних. Метод pct_change() обчислює відносну зміну між поточним і попередним значеннями в датафреймі або серії. Для кожного елемента вхідного об'єкта він обчислює, на скільки відсотків змінився поточний елемент порівняно з попереднім елементом. Цей метод є особливо корисним при аналізі фінансових даних, таких як ціни акцій, де часто цікаво визначити відсоткову зміну значень з часом. Результатом є новий об'єкт з відносними змінами, де значення представлені у відсотках.

3.4.5. Отримання прогнозів після використання методу опорних векторів

Нарешті, ми можемо генерувати прогнози, використовуючи модель як для набору даних на яких проводилося тренування моделі, так і для тестових даних, щоб отримати візуальну індикацію навичок моделі. Детальніше див. Додаток Б.

```
76  mse = mean_squared_error(testX,testY)*100
77
78  print('Test Score: %.2f MSE (%.2f RMSE)' % (mse, math.sqrt(mse)))
79
80  plt.plot(df['Cum_Ret'],color='red')
81  plt.plot(df['Cum_Strategy'],color='blue')
82  plt.show()
```

Для обрахування середньоквадратичної похибки скористаймося відповідною функцією з модулю sklearn.metrics, а саме mean_squared_error(). Оскільки результати вже у відсотках, то не має необхідності у додаткових обрахунках. На рисунку 3.5 можна побачити відсоток відхилення на навчальних і на тестових даних. Червона лінія на графіку – це реальні зміни, синя – прогнозований відсоток змін. Графіки інших компаній можна подивитись у додатку В.

Рисунок 3.5 – Результати дослідження для компанії EPAM методом опорних векторів

3.4.6. Оцінка якості методу опорних векторів

Оцінка якості методу опорних векторів включає в себе різні метрики та методи, які допомагають оцінити ефективність моделі. Декілька з таких метрик використовуються для оцінки точності та помилок моделі.

Одна з метрик - це середня помилка, яка визначається як середнє значення абсолютної відносної помилки між фактичними та прогнозованими значеннями. Для методу опорних векторів, взявши квадратний корінь з цієї помилки, ми можемо отримати загальну середню помилку моделі. Наприклад, якщо середня помилка складає 55.32%, то квадратний корінь з цього значення буде приблизно 0.086, що вказує на середню помилку на рівні 7.44%.

```
Test Score: 55.32 MSE (7.44 RMSE)
```

Точність моделі є ще однією важливою метрикою оцінки. Вона визначає відношення правильно класифікованих зразків до загальної кількості зразків. В нашому випадку можна зрозуміти, шо точність склала 92.56%.

Ці дві метрики - середня помилка та точність - дають загальне розуміння про ефективність моделі методу опорних векторів. Середня помилка вказує на середню відстань між прогнозованими та фактичними значеннями, виражену у відсотках. Точність визначає, яку частку зразків модель класифікує правильно. Отримані дані можна оцінити як успішну роботу методу опорних векторів.

3.4.7. Порівняння результатів роботи нейронних мереж та методу опорних векторів

Перевірка якості методів машинного навчання, включаючи нейронні мережі (ANN) та метод опорних векторів (SVM), за допомогою середньоквадратичної помилки (СКП) (mean squared error, MSE) є одним з популярних підходів для оцінки точності моделей.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.8}$$

СКП є мірою якості оцінювача. Оскільки вона походить від квадрата евклідової відстані, її значення є завжди додатним, і зменшується, коли похибка наближається до нуля.

Після тренування моделей використовуйте тестовий набір даних для прогнозування вихідних значень. Порівняйте прогнозовані значення з фактичними значеннями та обчисліть квадрат різниці між ними. MSE обчислюється як середнє значення квадратів всіх різниць між прогнозованими та фактичними значеннями. Щоб обчислити MSE, виконайте наступні кроки:

MSE дозволяє виміряти середньоквадратичну помилку моделі. Чим менше значення MSE, тим краще модель. Ви можете порівнювати значення MSE між різними моделями або використовувати його для порівняння моделей ANN та SVM. Модель з нижчим значенням MSE вважається точнішою і має меншу

середньоквадратичну помилку. Але для більш наочного відображення результатів оцінки використаємо RMSE, яка також є часто використовуваною[30].

RMSE (Root Mean Square Error) - це метрика, яка використовується для вимірювання відповідності між прогнозованими значеннями моделі і фактичними спостереженнями. Вона вимірює середньоквадратичну відстань між прогнозованими значеннями і дійсними значеннями. RMSE використовується для оцінки точності моделей прогнозування, де менші значення RMSE вказують на кращу точність моделі. Ця метрика штрафує великі відхилення прогнозів від дійсних значень, тому вона відображає загальну помилку моделі.

У наведеній таблиці результати перевірки RMSE для моделей SVM і ANN показують середнє значення RMSE для кожної компанії. Це значення вказує на середню помилку прогнозів кожної моделі для даної компанії.

Таблиця 4.1.

Результати порівняння RMSE для моделей SVM і ANN

COMPANY	**SVM**	**ANN**
EPAM	7.44%	2.4%
ADBE	46.51%	2.1%
CSCO	20.47%	2.7%
GLOB	5.67%	3.3%
GOOG	4.39%	2.5%
IBM	9.49%	2%
META	6.61%	3%
MSFT	42.44%	2.2%
ORCL	48.06%	2.1%
VMW	15.15%	2.4%
Середнє значення	20.62%	2.4%

Результати перевірки RMSE для моделей SVM і ANN наведені в таблиці. RMSE вимірює розброс між прогнозованими значеннями моделі і фактичними значеннями. Чим менше значення RMSE, тим краще модель прогнозує дані. За результатами перевірки, можна зробити такі спостереження:

1. Модель SVM має значення RMSE в діапазоні від 4.39% до 48.06%, з середнім значенням 20.62%. Це означає, що середня помилка прогнозування моделі SVM становить 20.62%. Вона показує розброс прогнозів моделі відносно фактичних значень.
2. Модель ANN має значення RMSE в діапазоні від 2% до 3.3%, з середнім значенням 2.4%. Це означає, що середня помилка прогнозування моделі ANN становить 2.4%. Це значення нижче, ніж у моделі SVM, що вказує на те, що модель ANN має менший розброс прогнозів і вважається точнішою для даної задачі прогнозування.
3. Взагалі, модель ANN показує кращу точність у порівнянні з моделлю SVM, оскільки її значення RMSE нижче.

Ці результати свідчать про те, що модель ANN є більш ефективною для прогнозування цін акцій на фондовому ринку, оскільки вона має меншу середню помилку прогнозування в порівнянні з моделлю SVM. Однак, важливо враховувати, що на практиці оцінка якості моделей може базуватися не тільки на RMSE, але і на інших метриках та контексті задачі.

РОЗДІЛ 4. ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ КОТИРУВАНЬ ФОНДОВОГО РИНКУ

4.1. Створення системи розсилки сповіщень користувачам

Щоб підвищити зручність використання та доступність розробленої моделі машинного навчання для аналізу та прогнозування фондового ринку, була реалізована система розсилки, яка забезпечує своєчасне сповіщення користувачів. Система розсилки була розроблена для надсилання сповіщень та оновлень підписаним користувачам на основі конкретних критеріїв і подій.

Основною метою системи розсилки є інформування користувачів про значні зміни на фондовому ринку, такі як раптові коливання цін, ринкові тенденції або виконання певних заздалегідь визначених умов. Ця функція дозволяє користувачам залишатися в курсі новин без необхідності постійного моніторингу ринку вручну.

У якості кінцевих користувачів були обрані менеджери та аналітики компанії які мають безпосереднє відношення до проекту збору та аналізу даних з аналітикою ситуації на біржі, а також розробники команди. Оскільки у якості корпоративної пошти використовується Microsoft Outlook це значно спростить завдання. На кожному з серверів, які використовуються на проекті, вже авторизований авто-користувач – обліковий запис, який використовується для автоматичної генерації та розсилки повідомлень.

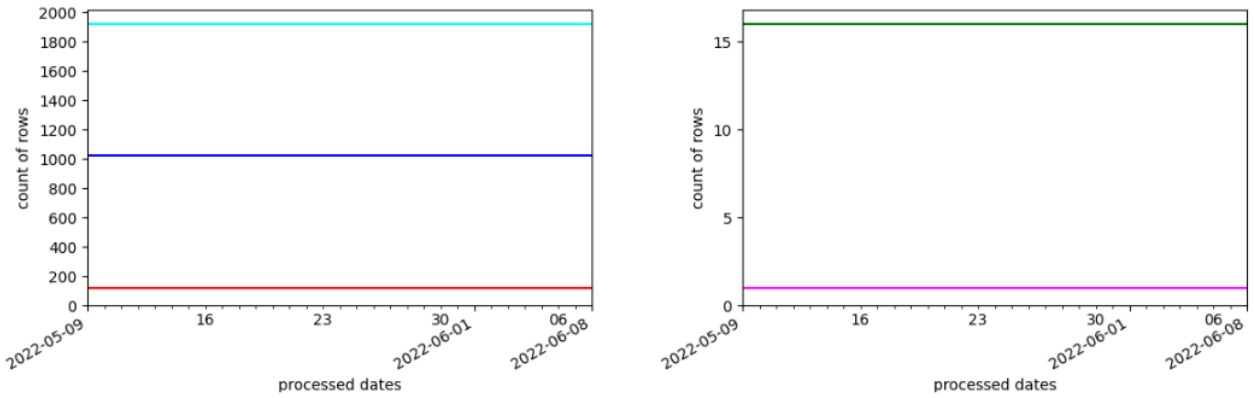


Рисунок 4.1. Приклад щоденної розсилки для підтвердження якості даних.

Залишається лише інтегрувати отримані графіки у систему розсилки і після узгодження з кінцевими користувачами налаштувати розсилку починати відправляти електронні листи.

Використовуючи систему розсилки, користувачі можуть налаштовувати свої вподобання та отримувати сповіщення електронною поштою чи іншими бажаними каналами зв'язку. Система збирає відповідні дані з моделі машинного навчання та надсилає відповідні сповіщення, що дозволяє користувачам швидко приймати обґрунтовані рішення.

4.2. Регулярний збір даних від Yahoo Finance

Щоб забезпечити точність і надійність моделі машинного навчання, було реалізовано надійний механізм збору даних із використанням платформи Yahoo Finance. Yahoo Finance надає широкий спектр фінансових даних, включаючи історичні ціни акцій, інформацію про компанії та ринкові новини, що робить його ідеальним джерелом для збору відповідних даних для аналізу та прогнозування.

Процес збору даних передбачає регулярне отримання даних фондового ринку, таких як ціни відкриття та закриття, обсяги торгів та інші відповідні показники, з інтерфейсу прикладного програмування (API) Yahoo Finance. Для цього знов використаємо мову програмування python. Щоб зібрати дані з Yahoo Finance за допомогою мови програмування Python, використовувалися різні бібліотеки та методи.

Процес збору даних починається з використання можливостей Python для взаємодії з веб-інтерфейсами API та збирання даних із веб-сайтів. У цьому випадку було здійснено доступ до платформи Yahoo Finance для отримання потрібної фінансової інформації.

Процес передбачав використання бібліотеки "request" для надсилання HTTP-запитів до API Yahoo Finance та бібліотеки "BeautifulSoup" для аналізу та вилучення відповідних даних із повернутого вмісту HTML. Ці бібліотеки дозволяють

отримувати різні точки даних, включаючи історичні ціни акцій, обсяги торгів та інші ринкові показники.

Для забезпечення ефективного та систематичного збору даних було реалізовано механізм циклу для повторення протягом заданого періоду часу або списку цільових запасів. Для кожної ітерації сценарій Python створював відповідну URL-адресу запиту API, включаючи бажані параметри, такі як біржовий символ і діапазон дат.

Отримавши відповідь API, сценарій проаналізував вміст HTML за допомогою BeautifulSoup, витягнувши необхідні поля даних і зберігаючи їх у відповідній структурі даних. Це може бути pandas DataFrame або користувацька структура даних, розроблена для відповідності конкретним потребам моделі аналізу чи прогнозування.

Для обробки сценаріїв, коли дані недоступні, або для усунення можливих помилок було реалізовано методи обробки помилок. Ці методи включали перевірку дійсності даних, обробку винятків і впровадження механізмів для повторних запитів або реєстрації помилок для подальшого дослідження.

Крім того, процес збору даних може включати додаткові функціональні можливості, такі як фільтрація даних на основі попередньо визначених критеріїв або застосування методів перетворення даних для забезпечення сумісності з наступними етапами аналізу чи прогнозування.

Загалом, використовуючи можливості програмування на Python, процес збору даних від Yahoo Finance передбачав надсилання запитів API, розбір HTML-вмісту, вилучення відповідних точок даних і їх зберігання у відповідних структурах даних. Виконуючи сценарій збору даних, користувачі мали змогу систематично збирати необхідну фінансову інформацію, необхідну для подальшого аналізу та прогнозування.

Використовуючи велике сховище фінансових даних Yahoo Finance, модель машинного навчання оснащена актуальною та вичерпною інформацією,

необхідною для створення точних прогнозів і цінної інформації для котирувань фондового ринку. Немає необхідності розгортати окрему базу для нових даних. Отримані дані отмаються і обробляються батчем, а історичні дані зберігаються у вже розгорнутій раніше базі.

4.3. Застосування розробленої моделі на комерційних підприємствах

Розроблена модель машинного навчання для аналізу та прогнозування фондового ринку має значний потенціал для застосування в комерційних підприємствах. Інтегруючи модель в існуючі торгові системи або інвестиційні платформи, компанії можуть покращити процеси прийняття рішень і оптимізувати інвестиційні стратегії.

Комерційні підприємства можуть використовувати цю модель для аналізу історичних тенденцій, виявлення закономірностей і робити обґрунтовані прогнози щодо майбутніх котирувань на фондовому ринку. Це дозволяє підприємствам отримати конкурентну перевагу за рахунок ефективного розподілу ресурсів, управління ризиками та максимізації віддачі від інвестицій.

Крім того, модель може допомогти в управлінні портфелем, надаючи інформацію в режимі реального часу та пропозиції щодо перебалансування або коригування інвестиційних портфелів на основі мінливих ринкових умов. Використовуючи модель машинного навчання, комерційні підприємства можуть приймати рішення на основі даних, які відповідають їхнім інвестиційним цілям і покращують загальну фінансову ефективність.

Застосування моделі машинного навчання на комерційних підприємствах також може сприяти автоматизації процесів. Модель може бути інтегрована в автоматичні системи торгівлі, де вона може самостійно аналізувати ринкові дані, генерувати прогнози та надсилати сигнали для здійснення угод. Це дозволяє підприємствам зосередитися на стратегічних аспектах торгівлі та зменшити залежність від ручного аналізу та прийняття рішень.

Окрім застосування моделі на комерційних підприємствах, її потенціал розширюється і на інші сфери фінансової діяльності. Наприклад, фондові брокери та інвестиційні консультанти можуть використовувати модель для надання рекомендацій клієнтам щодо інвестицій, розподілу активів та ризикового профілю. Фінансові аналітики можуть використовувати модель для дослідження ринків, проведення аналізу ризиків та виявлення нових інвестиційних можливостей.

Продовження досліджень і розвиток проекту можуть також сприяти створенню спеціалізованих інструментів та платформ, які використовують модель машинного навчання. Це можуть бути онлайн-сервіси, додатки або платформи, які надають користувачам доступ до аналізу ринку, прогнозування котирувань та порад щодо інвестицій. Розвиток таких інструментів може сприяти поширенню застосування моделі машинного навчання та демократизації доступу до фінансової аналітики та прогнозування на ринку акцій.

В цілому, застосування розробленої моделі машинного навчання на комерційних підприємствах має значний потенціал для покращення процесів прийняття рішень, оптимізації інвестиційних стратегій та покращення фінансових результатів. Подальші дослідження та розвиток проекту можуть розширити його застосування на інших ринках та фінансових секторах, а також сприяти створенню спеціалізованих інструментів та платформ для широкого використання моделі машинного навчання у фінансовій сфері.

4.4. Потенціал розвитку досліджень і реалізацій проекту

Застосування методів машинного навчання для аналізу та прогнозування біржових котирувань має значний потенціал розвитку. Проект уже продемонстрував свою цінність завдяки впровадженню системи розсилки для сповіщень користувачів і щоденного збору даних від Yahoo Finance. Однак є ще кілька шляхів для подальшого розвитку та вдосконалення.

Однією з сфер із великим потенціалом для просування є інтеграція додаткових джерел даних. Хоча Yahoo Finance надає велику кількість фінансових даних, включення даних з інших джерел може розширити можливості моделі. Наприклад, інтеграція даних із фінансових новин, платформ соціальних медіа або альтернативних постачальників даних може забезпечити більш повне уявлення про динаміку ринку та потенційно підвищити точність прогнозів.

Крім того, сама модель може бути додатково вдосконалена та оптимізована для підвищення її продуктивності. Постійні дослідження та розробки можуть досліджувати передові алгоритми машинного навчання, методи ансамблю або архітектури глибокого навчання. Ці підходи можуть допомогти охопити складні закономірності та залежності в даних фондового ринку, що призведе до більш точних прогнозів і цінної інформації.

Крім того, включення методів пояснюваного штучного інтелекту (XAI) може покращити інтерпретативність прогнозів моделі. Це надзвичайно важливо в сфері фондового ринку, де розуміння обґрунтування прогнозів має важливе значення для зміцнення довіри та забезпечення дотримання нормативних вимог. Надаючи прозорі пояснення, користувачі можуть отримати уявлення про фактори, що впливають на прогнози моделі, дозволяючи їм приймати більш обґрунтовані рішення.

Крім того, проведення ретельних ретестів і порівняльних досліджень може дати цінну інформацію про ефективність моделі порівняно з традиційними підходами та іншими моделями машинного навчання. Систематично оцінюючи сильні та слабкі сторони моделі, дослідники та практики можуть визначити сфери, які потребують вдосконалення, та скеровувати подальші вдосконалення.

Етичні міркування також повинні залишатися в центрі уваги. Забезпечення чесності, прозорості та захисту конфіденційності під час розробки та розгортання моделі є надзвичайно важливим. Це передбачає ретельне тестування на

упередженість, впровадження механізмів для вирішення потенційних несправедливих переваг і дотримання відповідних норм захисту даних.

Майбутній напрямок проекту також передбачає пошук партнерства з комерційними підприємствами. Співпрацюючи з підприємствами фінансової індустрії, розроблену модель можна розгорнути в реальних торгових системах або інвестиційних платформах. Це дозволяє підприємствам використовувати інформацію та прогнози моделі для вдосконалення процесів прийняття рішень, оптимізації інвестиційних стратегій і покращення фінансових показників.

Підсумовуючи, потенціал розвитку проекту полягає в кількох сферах. Вони включають інтеграцію додаткових джерел даних, постійне вдосконалення та оптимізацію моделі машинного навчання, включення зрозумілих методів штучного інтелекту, сувору оцінку ефективності, дотримання етичних міркувань і встановлення партнерства з комерційними підприємствами. Дотримуючись цих напрямків, проект може розвиватися, щоб запропонувати ще точніші прогнози, цінну інформацію та ширше застосування в галузі аналізу та прогнозування фондового ринку.

ВИСНОВКИ

У цьому дослідженні було проведено аналіз і прогнозування котувань на фондовій біржі за допомогою методів машинного навчання. Дослідження розглядалося з точки зору фінансового ринку і прогнозування цінової динаміки акцій.

У першому розділі проведено докладний аналіз проблематики прогнозування фінансових котувань. Визначено задачі прогнозування фінансових котувань та розглянуто методи машинного навчання, які застосовуються для цих задач. Детально розглянуто постановку задачі аналізу і прогнозування котувань акцій на фондовому ринку з використанням методів машинного навчання. Також проведений аналіз актуальності застосування методів машинного навчання для прогнозування котувань акцій на фондовому ринку.

У другому розділі детально розглянуто математичну модель штучних нейронних мереж, включаючи штучний нейрон, активаційну функцію, багатошарові штучні нейронні мережі, навчання штучних нейронних мереж та алгоритми навчання. Також розглянуто метод опорних векторів, включаючи його математичну модель, недоліки та шляхи вдосконалення роботи методу.

У третьому розділі описано формування бази даних котувань акцій, вибір засобів для реалізації моделі (мова програмування Python, репозиторій GitHub, інтегроване середовище розробки VSCode, бібліотеки TensorFlow, Keras, Scikit-learn, Pandas і matplotlib), підготовку середовища та серверної частини для програмування штучної нейронної мережі та методу опорних векторів. Також описано ініціалізацію моделей нейронних мереж і методу опорних векторів, отримання прогнозів, оцінку якості моделей та порівняння результатів роботи нейронних мереж та методу опорних векторів.

У четвертому розділі описано створення системи розсилки сповіщень користувачам, регулярний збір даних від Yahoo Finance, застосування розробленої

моделі на комерційних підприємствах і виявлено потенціал розвитку досліджень і реалізації проекту.

В роботі проведений аналіз і прогнозування котувань на фондовій біржі методами машинного навчання з використанням алгоритмів штучних нейронних мереж (ANN) та методу опорних векторів (SVM). Отримані результати підтверджують, що ці методи можуть бути ефективними інструментами для аналізу та прогнозування динаміки котувань на фондовій біржі.

Застосування ANN дозволяє моделювати складні нелінійні залежності між вхідними параметрами та цінами акцій. Цей підхід дозволяє отримати гнучку модель, здатну адаптуватись до змін у ринкових умовах. SVM, з свого боку, добре працює з невеликою кількістю вхідних параметрів і може ефективно вирішувати задачі класифікації та регресії.

Аналіз результатів показав, що обидва методи дають прийнятні результати, але ANN, в окремих випадках виявився набагато точнішим у прогнозуванні котувань акцій. Це може бути пов'язано з його здатністю моделювати складні залежності та використовувати більшу кількість вхідних параметрів.

Отже, на основі отриманих результатів можна зробити висновок, що застосування методів машинного навчання, зокрема ANN і SVM, може бути корисним для аналізу і прогнозування котувань на фондовій біржі. Враховуючи певні обмеження та особливості кожного методу, дослідникам і трейдерам слід ретельно обирати підхід, який найкраще відповідає їх потребам і умовам ринку.

Зазначимо, що результати даного дослідження є орієнтовними і можуть бути покращені шляхом вдосконалення моделей та врахування додаткових факторів, що впливають на ринок фондових котирувань. В роботі були досліджені реальні дані. Методи машинного навчання допомогли в глибинному вивчені даних. Отримані результати можуть бути використані для подальшого дослідження і академічного розвитку теми. Крім того, важливо продовжувати дослідження в цьому напрямку з

метою поліпшення точності прогнозування і розширення застосування методів машинного навчання в фінансовому секторі.

Проте, слід зауважити, що точність прогнозування котувань на фондовій біржі завжди залишається в певній мірі непередбачуваною, оскільки ринок може піддається впливу багатьох факторів, включаючи політичні, економічні та соціальні події. Тому, необхідно продовжувати дослідження в цьому напрямку для поліпшення точності прогнозів та розширення області їх застосування в фінансовому секторі.

На мою думку розвиток роботи може бути цікавим з академічної точки зору. Розгляд інших методів машинного навчання, варіювання активаційних функцій та функцій втрат, включення до аналізу додаткових метрик чи обмежень, збільшення вибірки даних для аналізу і прогнозування може суттєво вплинути на результати роботи

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. French, Jordan (2017). "The time traveller's CAPM". Investment Analysts Journal. **46** (2): 81–96.
2. Informed decision/choice/judgment etc | meaning of informed decision/choice/judgment etc in Longman Dictionary of Contemporary English | LDOCE URL: https://www.ldoceonline.com/dictionary/informed-decision-choice-judgment-etc
3. Mahmud, Tahmida; Hasan, Mahmudul; Chakraborty, Anirban; Roy-Chowdhury, Amit (19 August 2016). A poisson process model for activity forecasting. 2016 IEEE International Conference on Image Processing (ICIP). IEEE. doi:10.1109/ICIP.2016.7532978.
4. Li, Rita Yi Man; Fong, Simon; Chong, Kyle Weng Sang (2017). "Forecasting the REITs and stock indices: Group Method of Data Handling Neural Network approach". Pacific Rim Property Research Journal. **23** (2): 123–160.
5. MBA 604 Business Forecasting Methods – Harry Kogetsidis URL: https://view.officeapps.live.com/op/view.aspx?src=http%3A%2F%2Fwww.mba.unic.ac.cy%2FMBA604%2FAveraging%2520and%2520Exponential%2520Smoothing%2520Methods.doc%23%3A~%3Atext%3Dmeasuring%2520forecast%2520accuracy.-%2CAveraging%2520methods%2Cmost%2520recent%2520n%2520time%2520periods.&wdOrigin=BROWSELINK
6. Sreekanth Menon, AI/ML leader, and Rajeev Ranjan, data science leader, Genpact. URL:https://www.genpact.com/insight/technical-paper/the-evolution-of-forecasting-techniques-traditional-versus-machine-learning-methods
7. R. C. Cavalcante, R. C. Brasileiro, V. L. Souza, J. P. Nobrega, and A. L. Oliveira, “Computational intelligence and financial markets: A survey and future directions,”

Expert Systems with Applications, vol. 55, pp. 194 – 211, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095741741630029X

8. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. **20** (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478.
9. A comparison between SVM and multilayer perceptron in predicting an emerging financial market: Colombian stock market. Oscar Bustos; Alexandra Pomares; Enrique Gonzalez. *URL:* https://ieeexplore.ieee.org/document/8273335 (дата звернення: 09.12.2021)
10. Новиков В.А. Калацкая Л.В., Садов В.С. Организация и обучение искусственных нейронных сетей: Экспериментальное учебное пособие Минск: БГУ, 2003. 72 с.
11. Nusrat Rouf, Majid Bashir Malik “Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions” 8 November 2021
12. Kerem Gülen “Applying machine learning in financial markets: A review of state-of-the-art methods” January 11, 2023 DataConomy *URL:* https://dataconomy.com/2023/01/11/stock-prediction-machine-learning/#:~:text=Long%20short%2Dterm%20memory%20(LSTM)%3A%20Many%20experts%20currently%20consider,promising%20algorithm%20for%20stock%20prediction. (дата звернення: 09.05.2023)
13. Naliniprava Tripathy “Stock Price Prediction Using Support Vector Machine Approach” (PDF) International Academic Conference on Management & Economy Oxford, United Kingdom 8-10 November 2019
14. Adam Hayes “Bollinger Bands: What They Are, and What They Tell Investors” January 18, 2023 Investopedia URL: https://www.investopedia.com/terms/b/bollingerbands.asp

15. Ming-Chi Lee “Using support vector machine with a hybrid feature selection method to the stock trend prediction” October 2009 URL: https://www.researchgate.net/publication/223855375_Using_support_vector_machine_with_a_hybrid_feature_selection_method_to_the_stock_trend_prediction
16. Justin Sirignano, Rama Cont “Universal features of price formation in financial markets: perspectives from deep learning” 09 Jul 2019 URL: https://doi.org/10.1080/14697688.2019.1622295
17. Хайкин С. Нейронные сети. Вильямс, 2006. – 1103 с.
18. Степанов В. А. Фондовый рынок и нейросети
19. Степанов В. А. Мир ПК. 1998. URL: http://www.osp.ru/pcworld/1998/12/159835
20. Герасименко Н. А. Нейросетевые технологии в анализе фондового рынка. 1998. URL: http://fakit.ru/main_dsp.php?top_id=1086
21. Kohonen T. Self-organization and associative memory. Series in Information Sciences, volume 8. Berlin: Springer Verlag. 1984.
22. Rosenblatt F. 1962. Principles of neurodynamics. New York: Spartan Books. (Російський переклад: Розенблатт Ф. Принципы нейродинамики. М.: Мир., 1965.)
23. Widrow B., Hoff M. Adaptive switching circuits. IRE WESCON Convention Record, pp. 96-104. New York: Institute of Radio Engineers. 1960.
24. Guido van Rossum, Python Reference Manual, release 2.4.4, 18 October 2006.
25. Martin Abadi, Ashish Agarwal “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems” November 9, 2015
26. Yahoo finance – stock market live quotes business & finance news URL: https://finance.yahoo.com/
27. Проніна, О. І. (2019). Робоча програма навчальної дисципліни «Машинне навчання» для здобувачів освітнього ступеня магістра за спеціальністю 122 «Комп’ютерні науки», освітньої програми «Інформаційні системи та технології»

28. Keras. The Sequential model URL: https://keras.io/guides/sequential_model/
29. scikit-learn: machine learning in Python URL: https://scikit-learn.org/stable/
30. Rajashree Dash, Pradipta K. Dash “A Hybrid FOREX Predictor Model Using a Legendre Polynomial Neural Network with a Modified Differential Harmony Search Technique Author links open overlay panel “ 21 July 2017 Handbook of Neural Computation 2017

ДОДАТКИ

Додаток А

TABLE IV
ACCURACY COMPARISON BETWEEN ANN AND SVN IN TEST DATA

Stock	ANN	SVM	Diff
BCOLOMBIA	0.74	0.78	-0.04
BOGOTA	0.65	0.73	-0.07
BVC	0.75	0.87	-0.12
CELSIA	0.81	0.74	0.07
CEMARGOS	0.82	0.82	0.00
CLH	0.75	0.77	-0.02
CNEC	0.79	0.75	0.05
CONCONCRET	0.78	0.78	0.00
CORFICOLCF	0.80	0.79	0.01
ECOPETROL	0.77	0.84	-0.07
EEB	0.73	0.81	-0.09
ETB	0.76	0.82	-0.06
EXITO	0.78	0.72	0.06
GRUPOARGOS	0.74	0.80	-0.06
GRUPOAVAL	0.72	0.78	-0.06
GRUPOSURA	0.75	0.77	-0.02
ISA	0.77	0.77	0.00
NUTRESA	0.75	0.76	-0.01
PFAVAL	0.76	0.72	0.04
PFAVH	0.71	0.77	-0.06
PFBCOLOM	0.80	0.75	0.05
PFCEMARGOS	0.76	0.79	-0.03
PFDAVVNDA	0.81	0.83	-0.03
PFGRUPOARG	0.75	0.86	-0.10
PFGRUPSURA	0.75	0.76	-0.01
Average	0.76	0.78	-0.02

ANN:

```
import pandas as pd
import datetime as dt
import numpy as np
import math
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from keras.models import Sequential
from keras.layers import Dense

# convert an array of values into a dataset matrix
def create_dataset(dataset, look_back=1):
	dataX, dataY = [], []
	for i in range(len(dataset)-look_back-1):
		a = dataset[i:(i+look_back), 0]
		dataX.append(a)
		dataY.append(dataset[i + look_back, 0])
	return np.array(dataX), np.array(dataY)

dataset = pd.read_excel("ChartData.xls", usecols=[4])
dataset = dataset.values
dataset = dataset.astype('float32')

dates = pd.read_excel("ChartData.xls", usecols=[0]).values
dates = dates.ravel()
x_dates = pd.to_datetime(dates, errors='ignore', format = '%m/%d/%Y')

# split into train and test sets
train_size = int(len(dataset) * 0.8)
test_size = len(dataset) - train_size
train = dataset[0:train_size,:]
test = dataset[train_size:len(dataset),:]
# reshape dataset
look_back = 3
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
# create and fit Multilayer Perceptron model
model = Sequential()
model.add(Dense(12, input_dim=look_back, activation='relu'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=2, verbose=2)
# Estimate model performance
```

```
trainScore = model.evaluate(trainX, trainY, verbose=0)
print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
testScore = model.evaluate(testX, testY, verbose=0)
print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))
# generate predictions for training
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
# shift train predictions for plotting
trainPredictPlot = np.empty_like(dataset)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(trainPredict) + look_back, :] = trainPredict
# shift test predictions for plotting
testPredictPlot = np.empty_like(dataset)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(trainPredict) + (look_back * 2) + 1:len(dataset) - 1,:] = testPredict
# plot baseline and predictions
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%d/%Y'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=98))
plt.plot(x_dates, dataset)
plt.plot(x_dates, trainPredictPlot)
plt.plot(x_dates, testPredictPlot)
plt.gcf().autofmt_xdate()
plt.show()
```

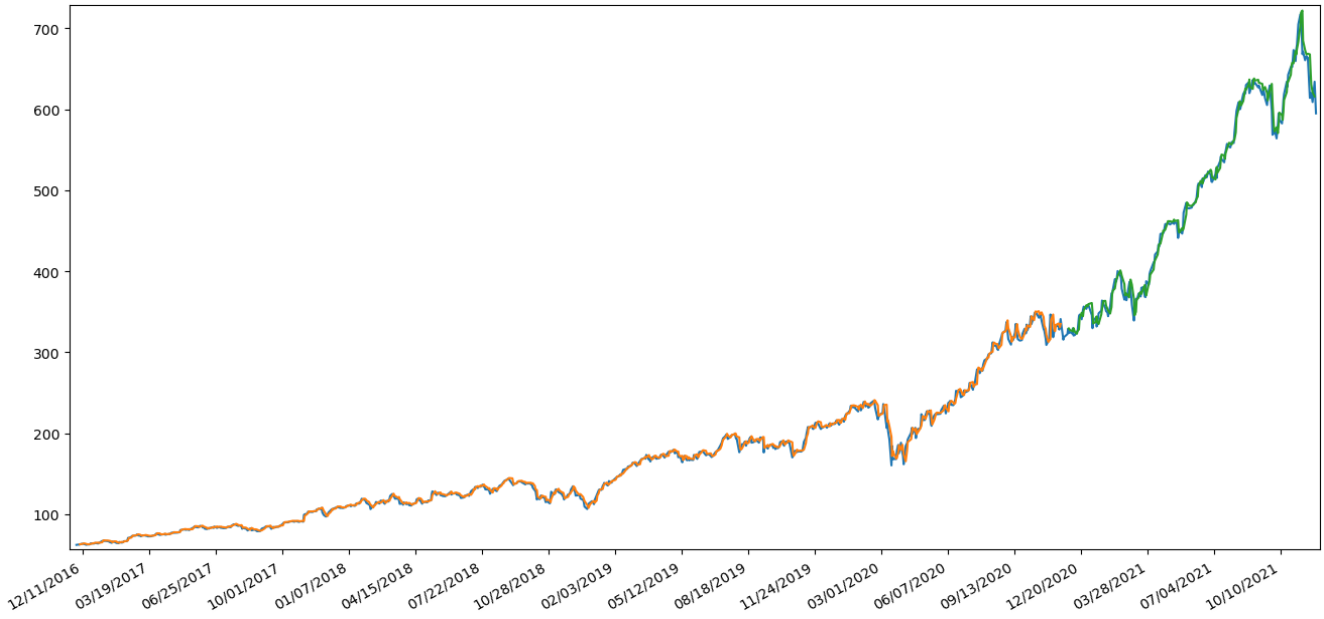
SVM:

```
from sklearn.svm import SVC
from sklearn.metrics import mean_squared_error
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
import warnings
warnings.filterwarnings("ignore")
import math
#df_EPAM = pd.read_csv('EPAM.csv')
#df_MSFT = pd.read_csv('MSFT.csv')
#df_ADBE = pd.read_csv('ADBE.csv')
#df_CSCO = pd.read_csv('CSCO.csv')
#df_GLOB = pd.read_csv('GLOB.csv')
#df_GOOG = pd.read_csv('GOOG.csv')
#df_IBM = pd.read_csv('IBM.csv')
#df_META = pd.read_csv('META.csv')
#df_ORCL = pd.read_csv('ORCL.csv')
df_VMW = pd.read_csv('VMW.csv')
```

```
df_VMW.index = pd.to_datetime(df_VMW['Date'])
df = df_VMW.drop(['Date'], axis='columns')
df['Open-Close'] = df.Open - df.Close
df['High-Low'] = df.High - df.Low
X = df[['Open-Close', 'High-Low']]

y = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)
split_percentage = 0.8
split = int(split_percentage*len(df))

# Train data set
X_train = X[:split]
y_train = y[:split]

# Test data set
X_test = X[split:]
y_test = y[split:]

# Support vector classifier
cls = SVC().fit(X_train, y_train)

df['Predicted_Signal'] = cls.predict(X)
# Calculate daily returns
df['Return'] = df.Close.pct_change()
# Calculate strategy returns
df['Strategy_Return'] = df.Return *df.Predicted_Signal.shift(1)

# Calculate Cumulutive returns
df['Cum_Ret'] = df['Return'].cumsum()
# Plot Strategy Cumulative returns
df['Cum_Strategy'] = df['Strategy_Return'].cumsum()

testX = df['Cum_Ret'].tail(len(df)-split).tolist()
testY = df['Cum_Strategy'].tail(len(df)-split).tolist()

mse = mean_squared_error(testX,testY)*100

print('Test Score: %.2f MSE (%.2f RMSE)' % (mse, math.sqrt(mse)))

plt.plot(df['Cum_Ret'],color='red')
plt.plot(df['Cum_Strategy'],color='blue')
plt.show()
```

EPAM:

ANN:

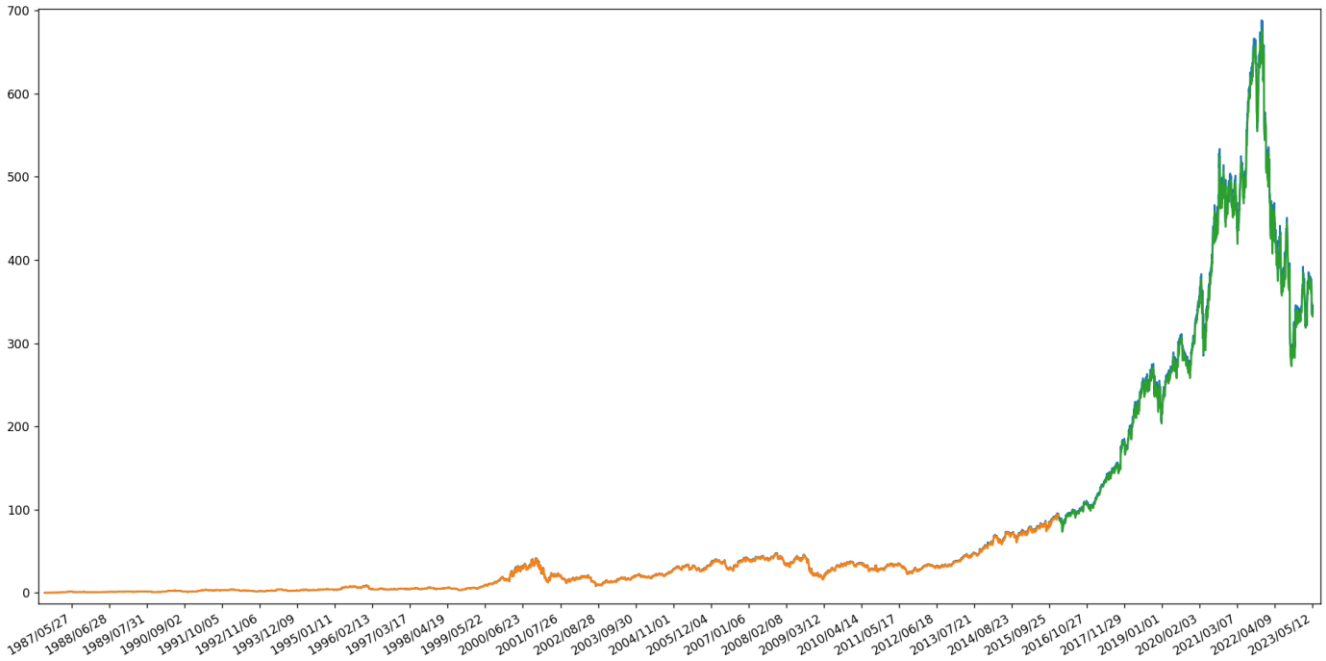

700
600
500
400
300
200
100
12/11/2016
03/19/2017
06/25/2017
10/01/2017
01/07/2018
04/15/2018
07/22/2018
10/28/2018
02/03/2019
05/12/2019
08/18/2019
11/24/2019
03/01/2020
06/07/2020
09/13/2020
12/20/2020
03/28/2021
07/04/2021
10/10/2021

SVM:

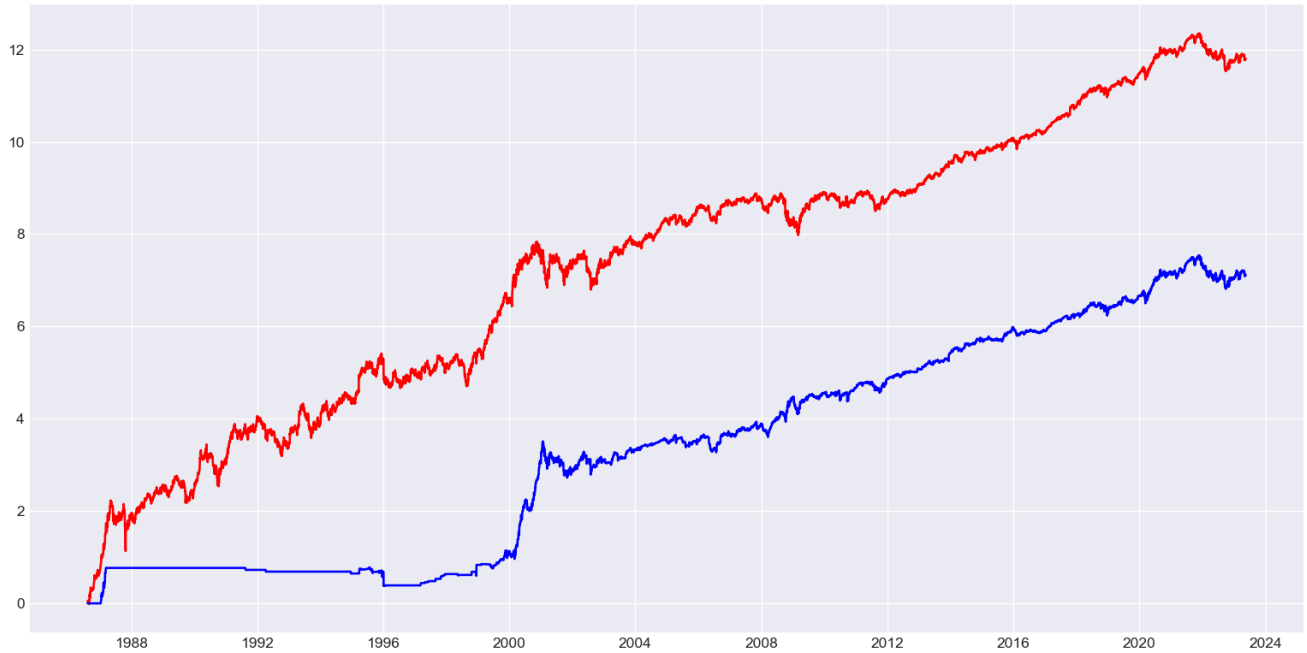

5
4
3
2
1
0
2012
2014
2016
2018
2020
2022

ADBE:

ANN:

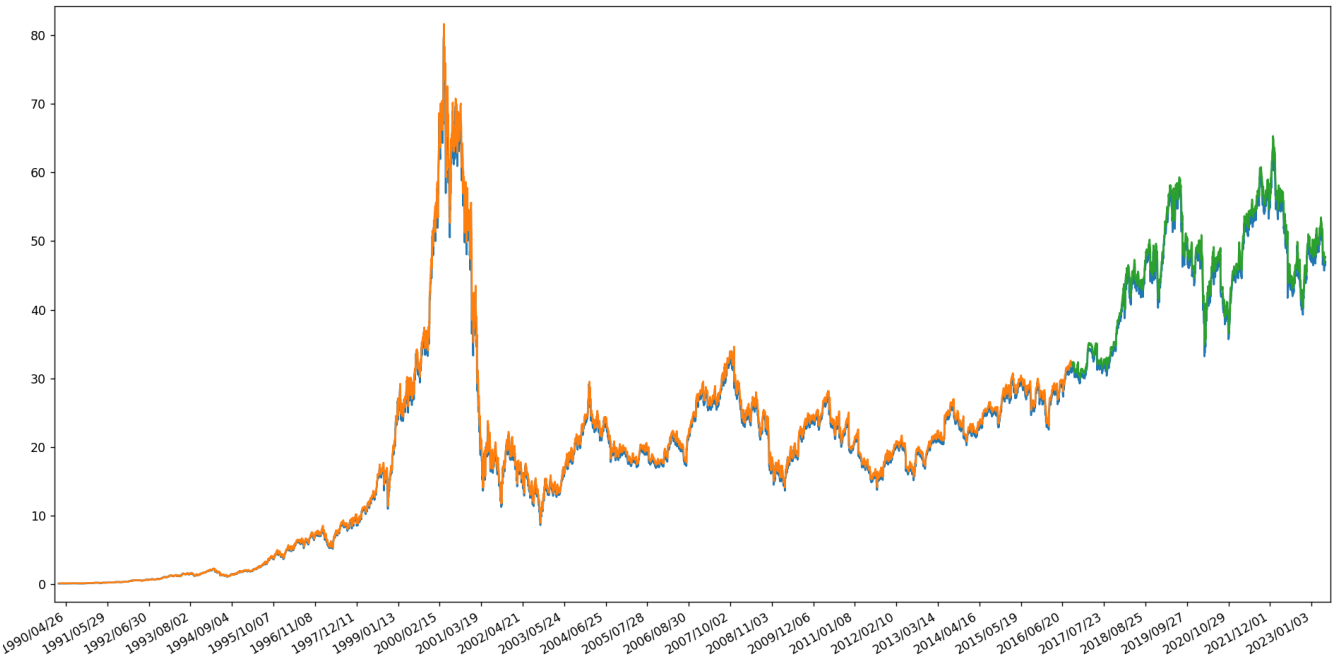

700
600
500
400
300
200
100
0
1987/05/27
1988/06/28
1989/07/31
1990/09/02
1991/10/05
1992/11/06
1993/12/09
1995/01/11
1996/02/13
1997/03/17
1998/04/19
1999/05/22
2000/06/23
2001/07/26
2002/08/28
2003/09/30
2004/11/01
2005/12/04
2007/01/06
2008/02/08
2009/03/12
2010/04/14
2011/05/17
2012/06/18
2013/07/21
2014/08/23
2015/09/25
2016/10/27
2017/11/29
2019/01/01
2020/02/03
2021/03/07
2022/04/09
2023/05/12

SVM:


12
10
8
6
4
2
0
1988
1992
1996
2000
2004
2008
2012
2016
2020
2024

CSCO:

ANN:


80
70
60
50
40
30
20
10
0
1990/04/26
1991/05/29
1992/06/30
1993/08/02
1994/09/04
1995/10/07
1996/11/08
1997/12/11
1999/01/13
2000/02/15
2001/03/19
2002/04/21
2003/05/24
2004/06/25
2005/07/28
2006/08/30
2007/10/02
2008/11/03
2009/12/06
2011/01/08
2012/02/10
2013/03/14
2014/04/16
2015/05/19
2016/06/20
2017/07/23
2018/08/25
2019/09/27
2020/10/29
2021/12/01
2023/01/03

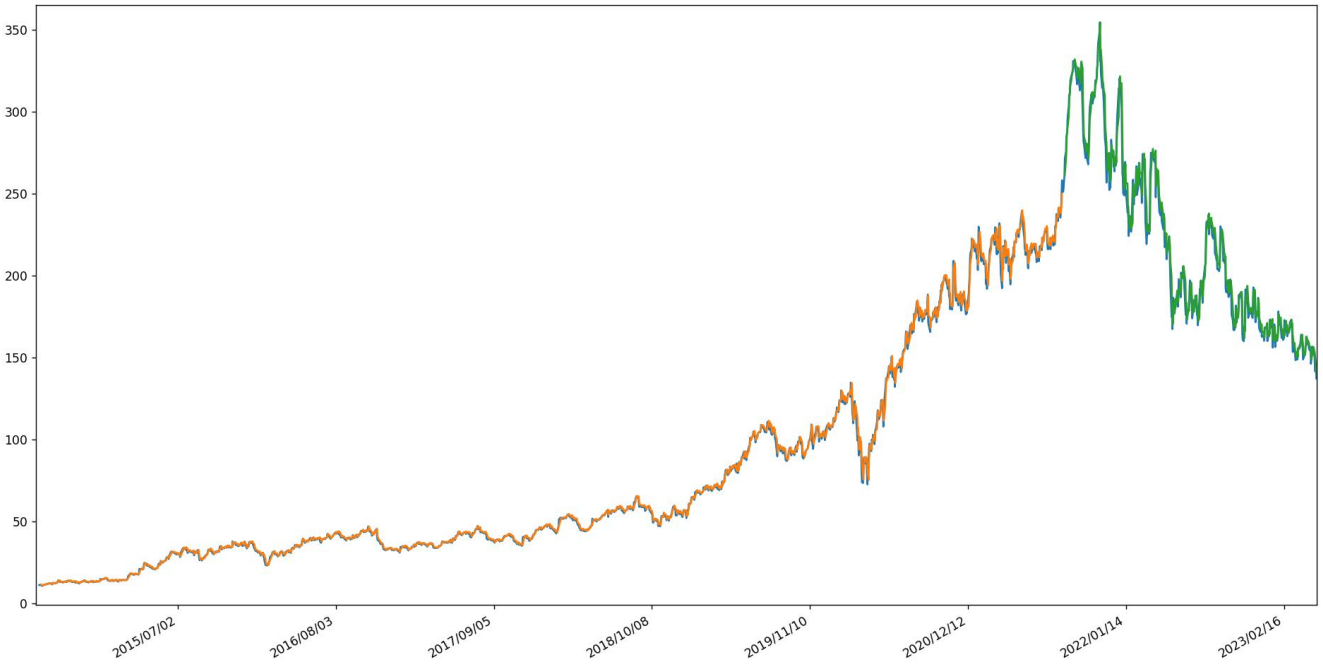
.

SVM:

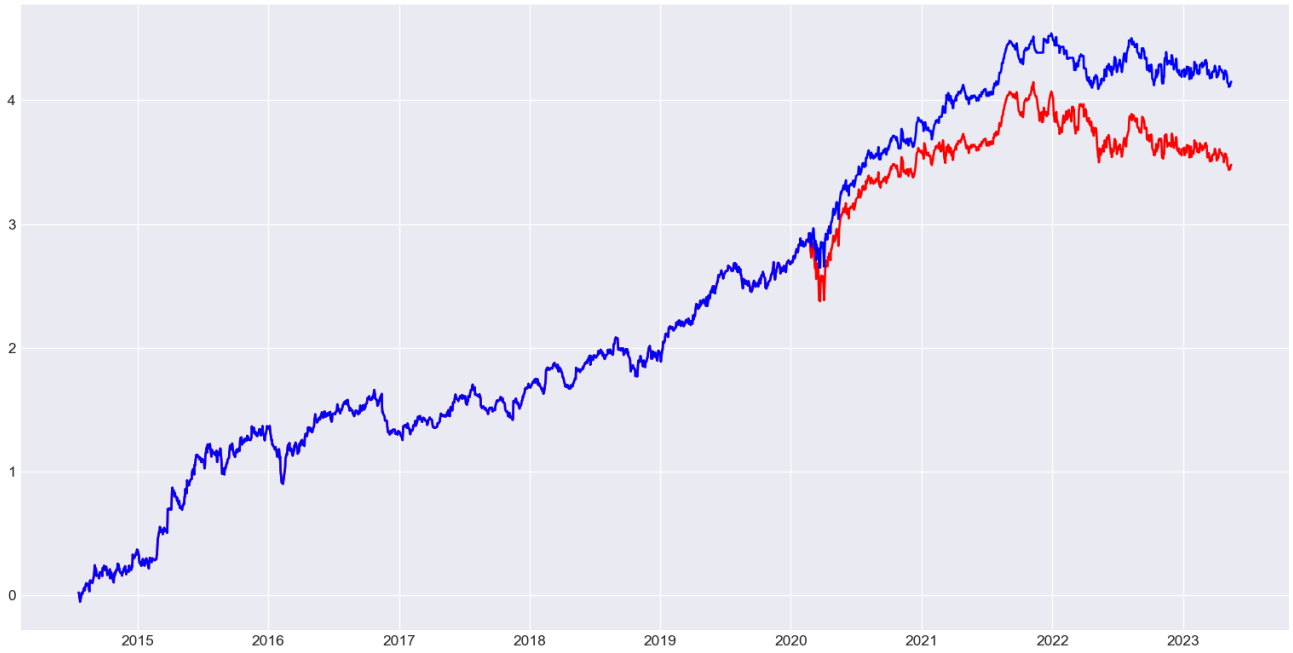

10
8
6
4
2
0
1992
1996
2000
2004
2008
2012
2016
2020
2024

GLOB:

ANN:

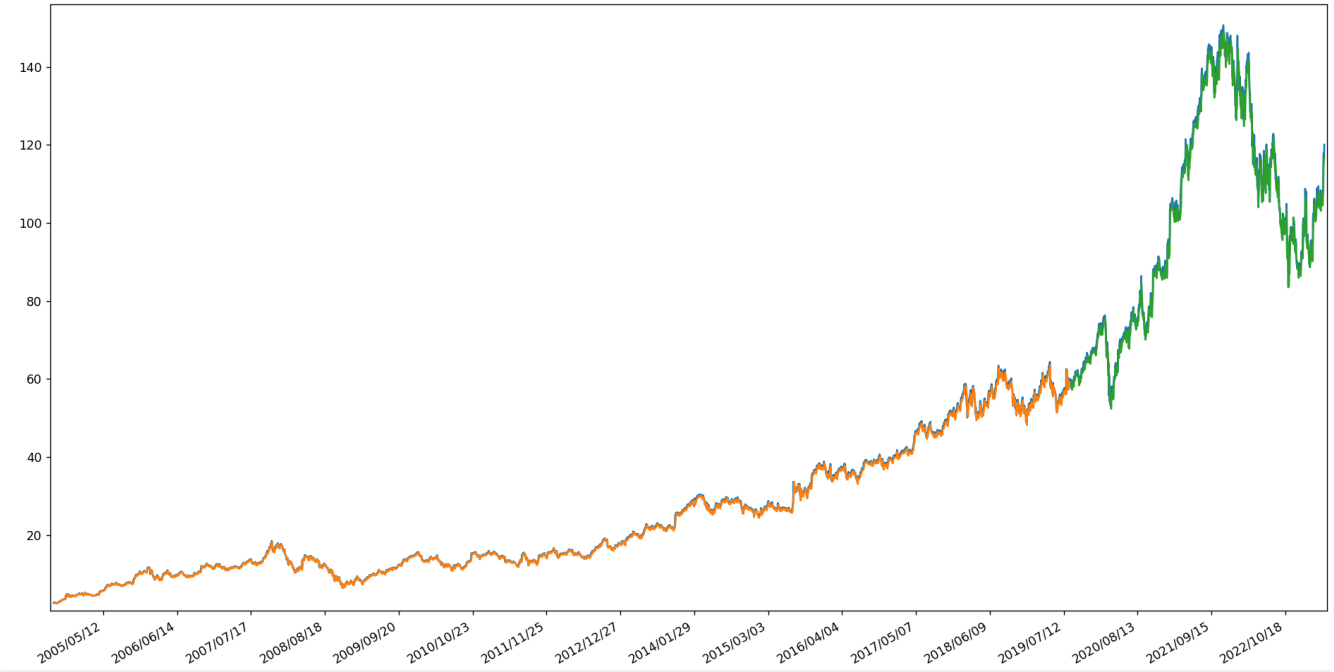

350
300
250
200
150
100
50
0
2015/07/02
2016/08/03
2017/09/05
2018/10/08
2019/11/10
2020/12/12
2022/01/14
2023/02/16

SVM:


4
3
2
1
0
2015
2016
2017
2018
2019
2020
2021
2022
2023

GOOG:

ANN:

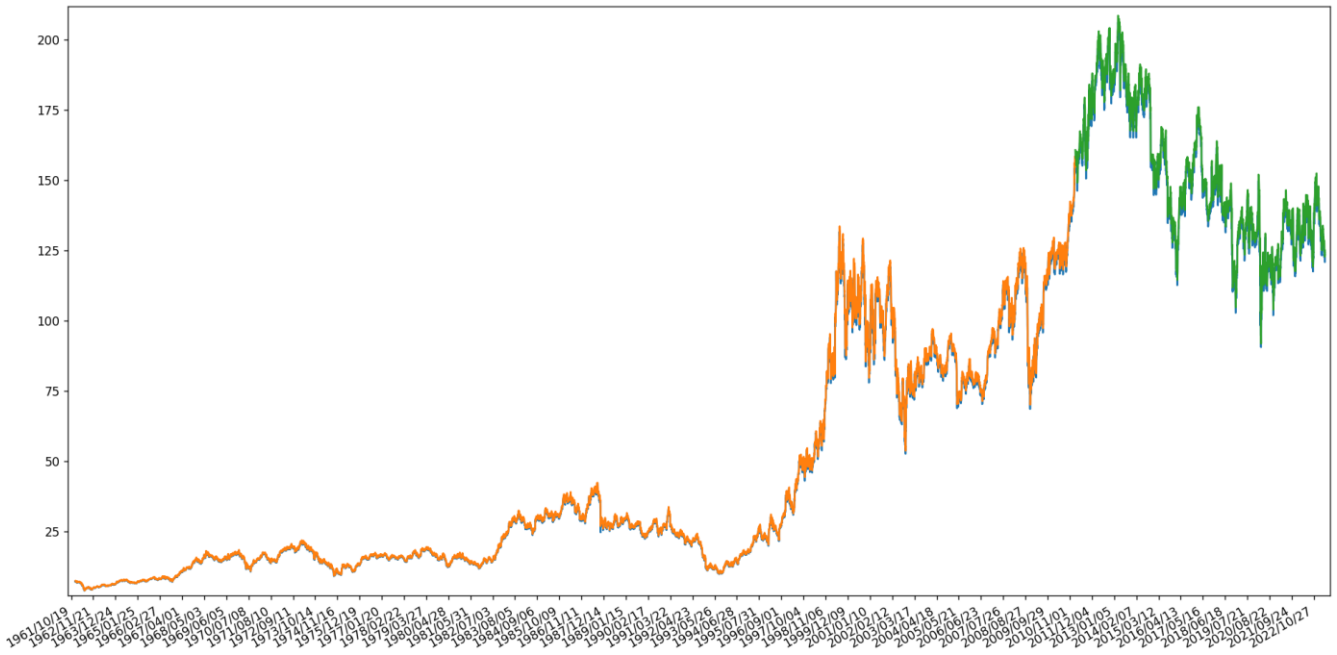

140
120
100
80
60
40
20
2005/05/12
2006/06/14
2007/07/17
2008/08/18
2009/09/20
2010/10/23
2011/11/25
2012/12/27
2014/01/29
2015/03/03
2016/04/04
2017/05/07
2018/06/09
2019/07/12
2020/08/13
2021/09/15
2022/10/18

SVM:

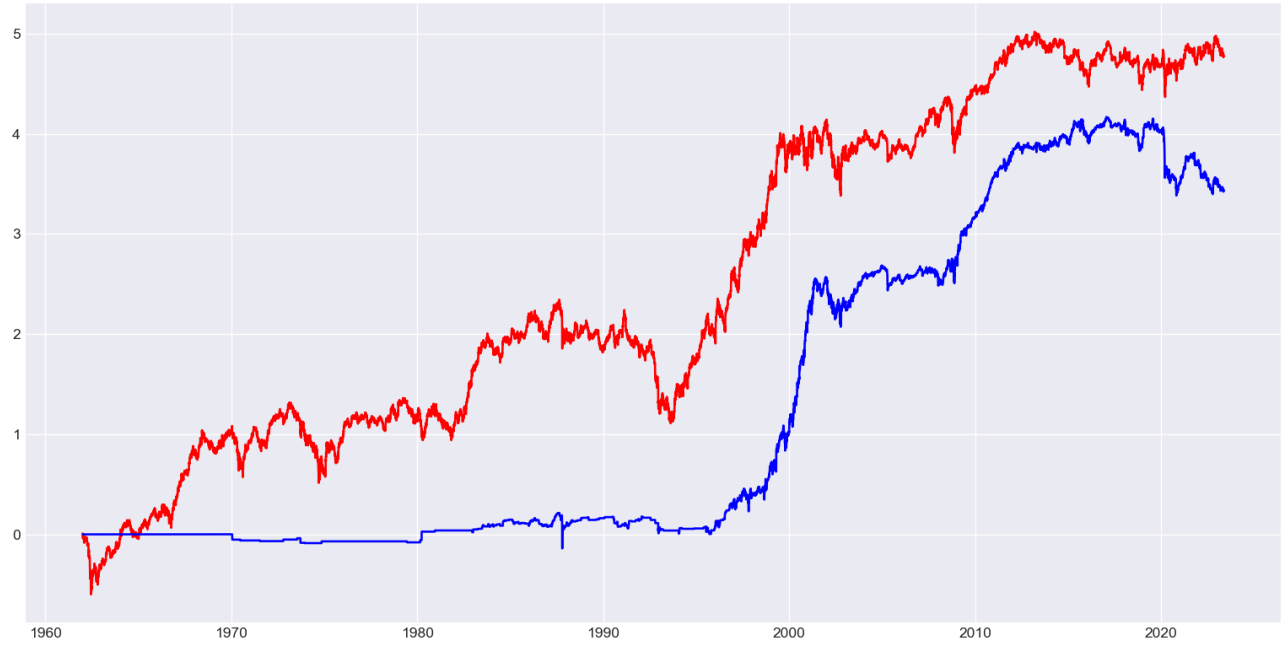

5
4
3
2
1
0
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024

IBM:

ANN:

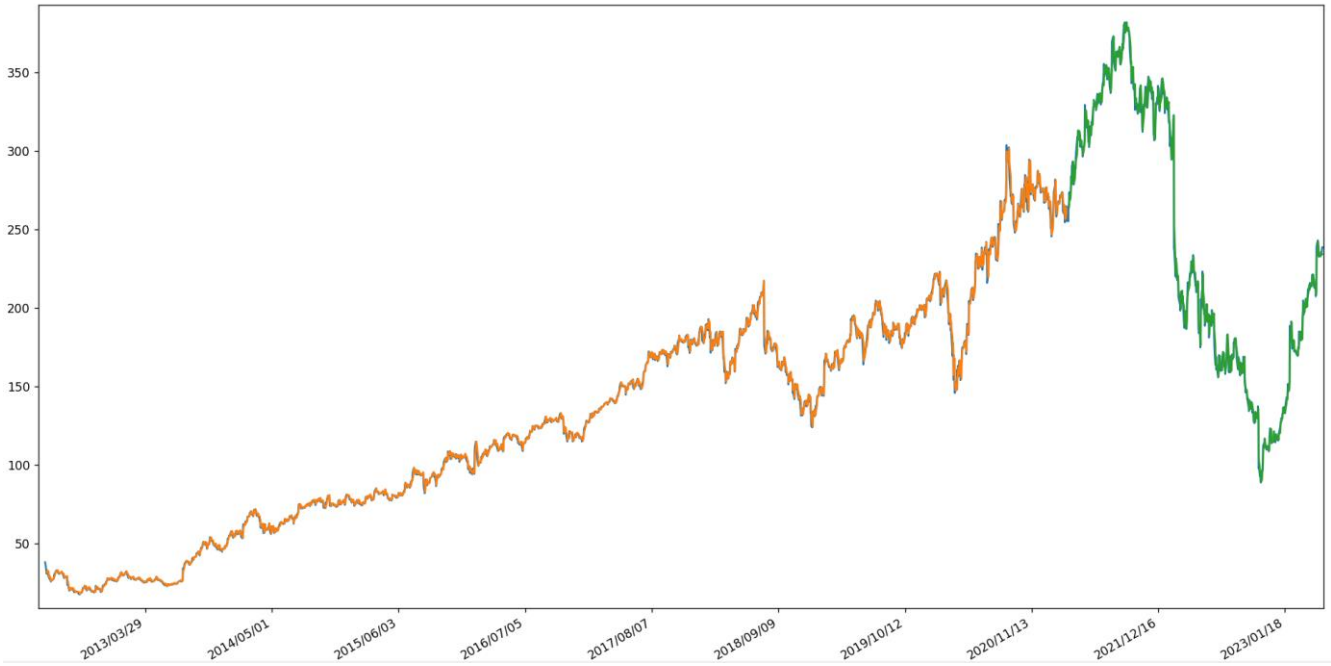

200
175
150
125
100
75
50
25

SVM:


5
4
3
2
1
0
1960
1970
1980
1990
2000
2010
2020

META:

ANN:

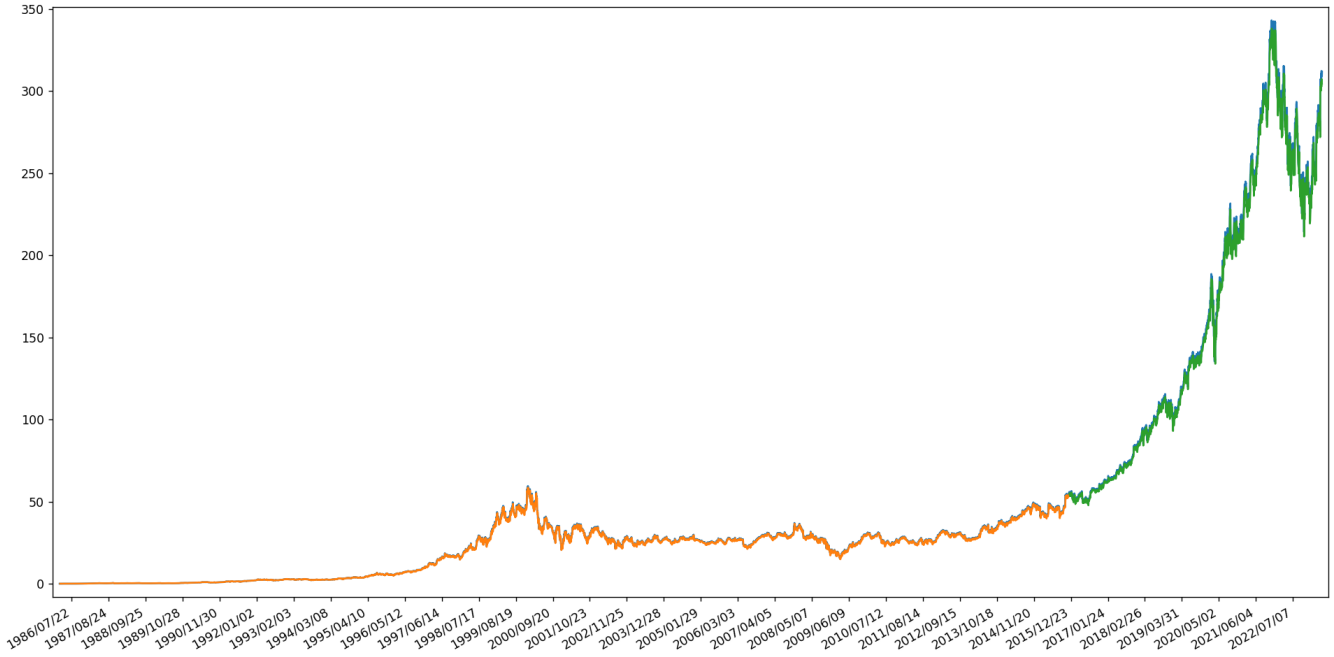

350
300
250
200
150
100
50
2013/03/29
2014/05/01
2015/06/03
2016/07/05
2017/08/07
2018/09/09
2019/10/12
2020/11/13
2021/12/16
2023/01/18

SVM:

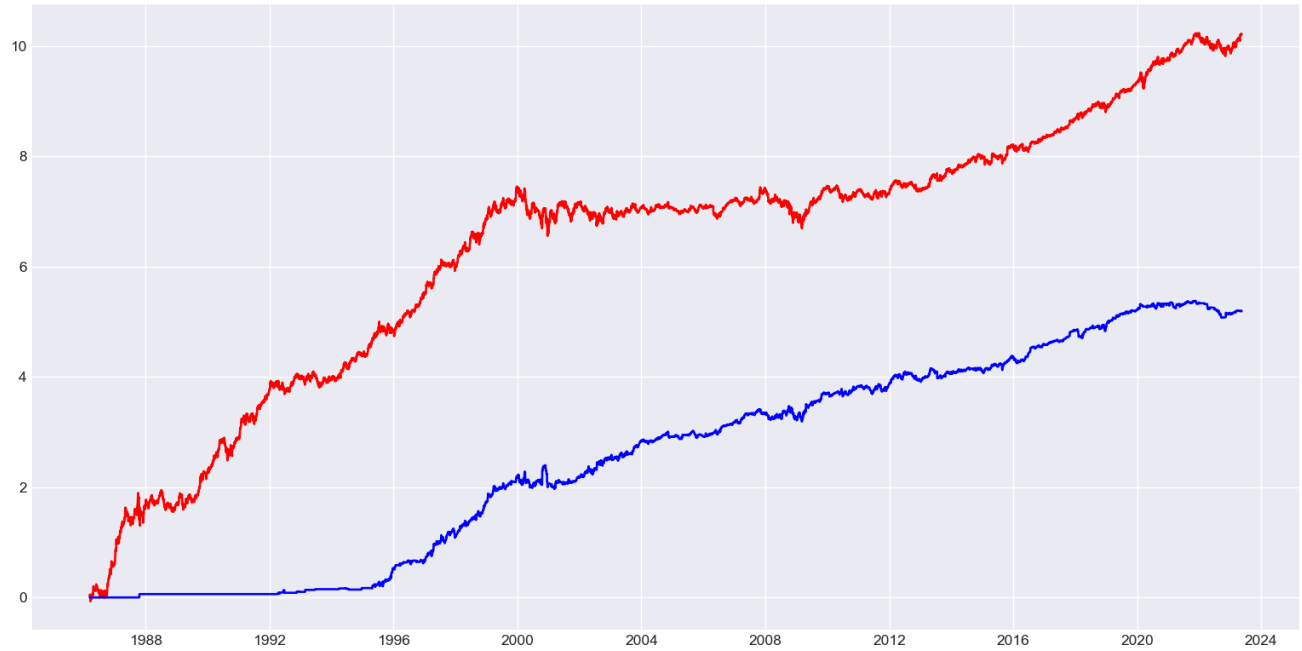

3
2
1
0
2012
2014
2016
2018
2020
2022

MSFT:

ANN:


350
300
250
200
150
100
50
0
1986/07/22
1987/08/24
1988/09/25
1989/10/28
1990/11/30
1992/01/02
1993/02/03
1994/03/08
1995/04/10
1996/05/12
1997/06/14
1998/07/17
1999/08/19
2000/09/20
2001/10/23
2002/11/25
2003/12/28
2005/01/29
2006/03/03
2007/04/05
2008/05/07
2009/06/09
2010/07/12
2011/08/14
2012/09/15
2013/10/18
2014/11/20
2015/12/23
2017/01/24
2018/02/26
2019/03/31
2020/05/02
2021/06/04
2022/07/07

SVM:

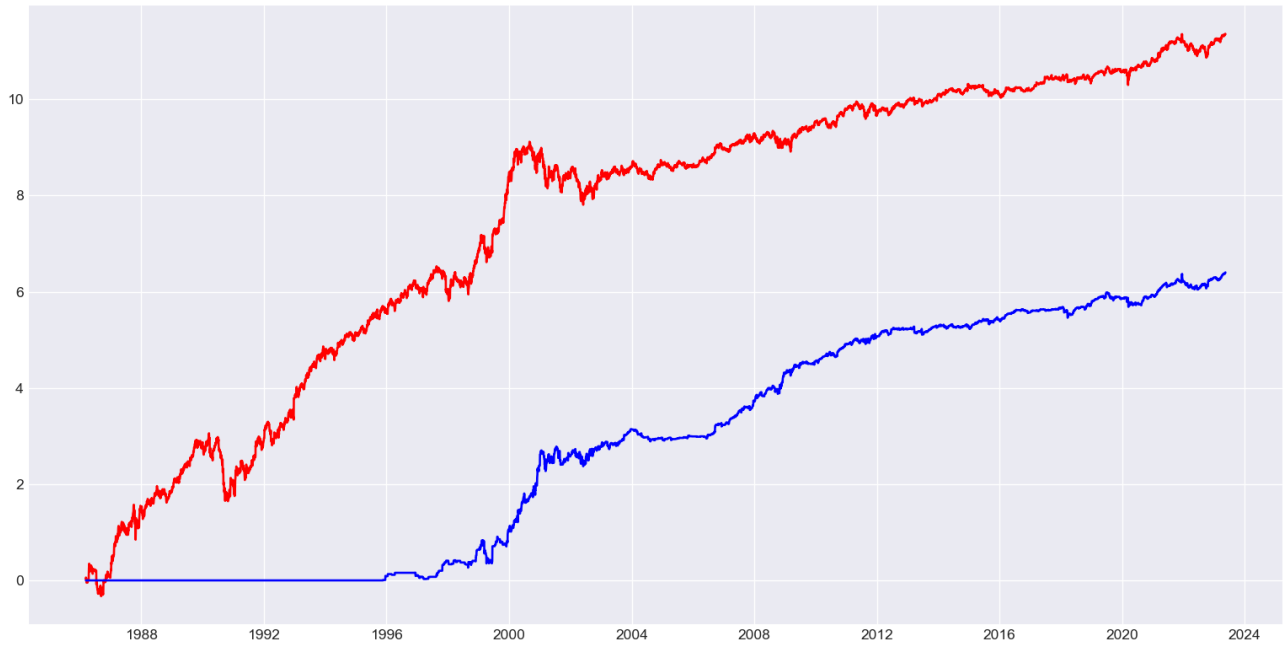

10
8
6
4
2
0
1988
1992
1996
2000
2004
2008
2012
2016
2020
2024

ORCL:

ANN:

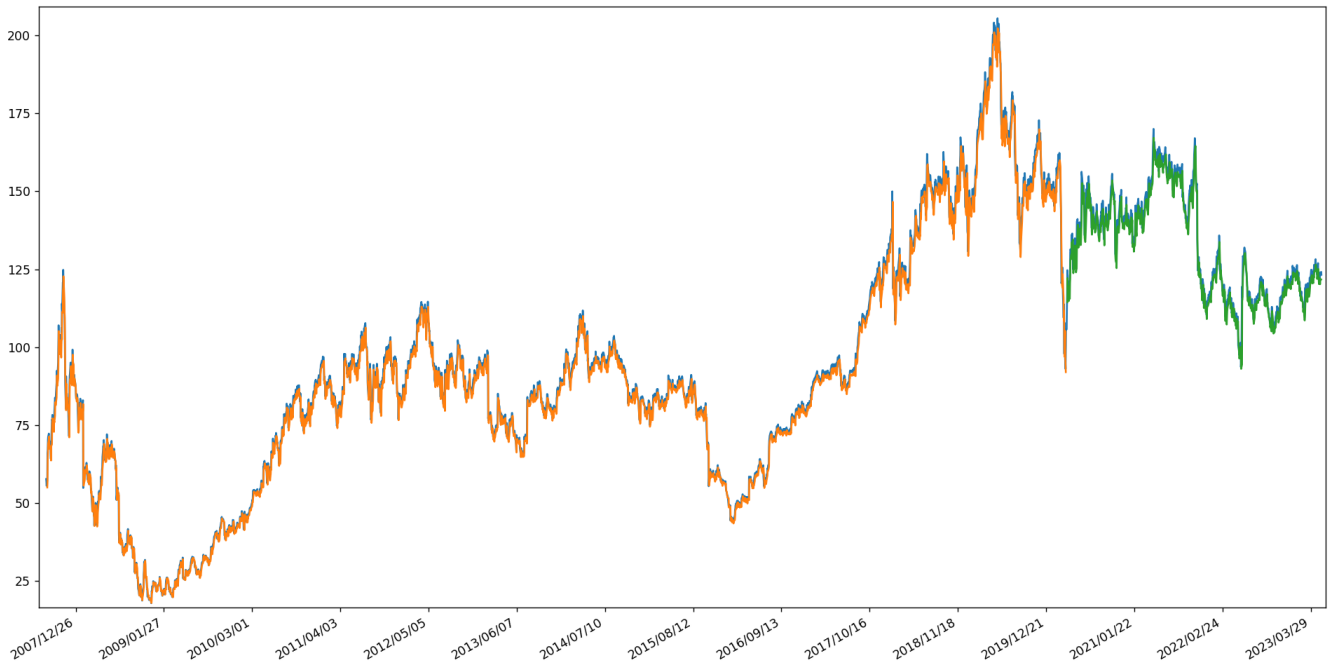

100
80
60
40
20
0
1986/10/17
1987/11/19
1988/12/21
1990/01/23
1991/02/25
1992/03/29
1993/05/01
1994/06/03
1995/07/06
1996/08/07
1997/09/09
1998/10/12
1999/11/14
2000/12/16
2002/01/18
2003/02/20
2004/03/24
2005/04/26
2006/05/29
2007/07/01
2008/08/02
2009/09/04
2010/10/07
2011/11/09
2012/12/11
2014/01/13
2015/02/15
2016/03/19
2017/04/21
2018/05/24
2019/06/26
2020/07/28
2021/08/30
2022/10/02

SVM:

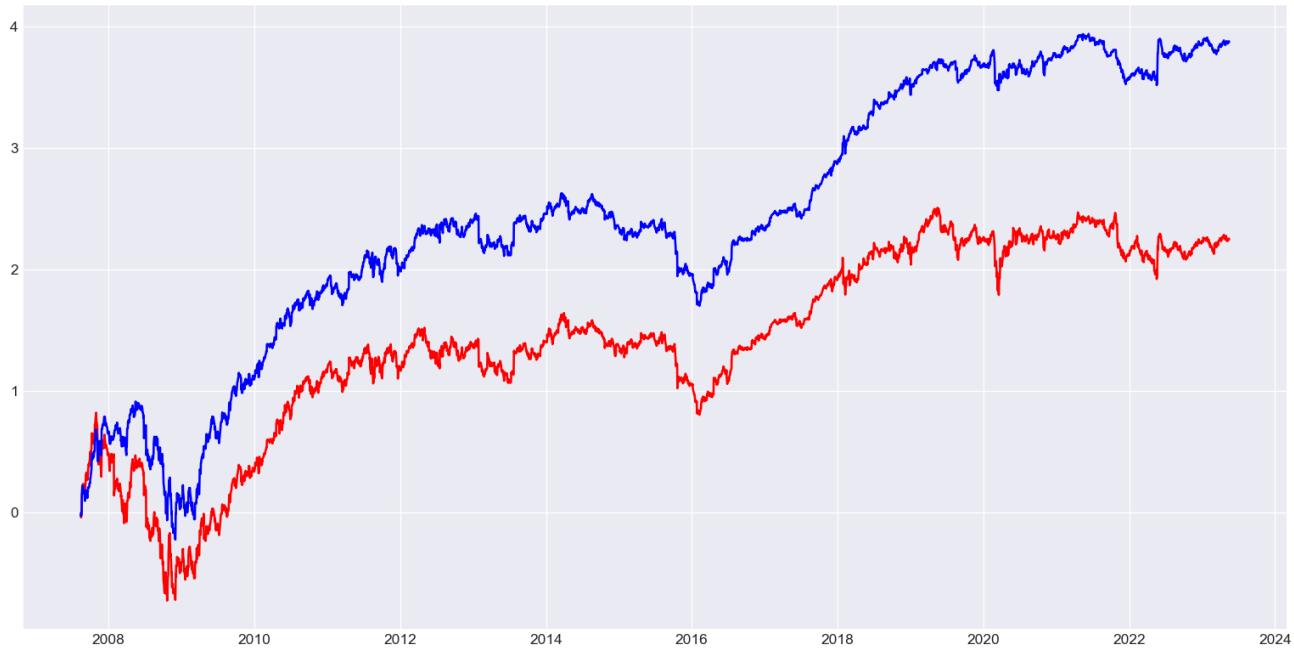

10
8
6
4
2
0
1988
1992
1996
2000
2004
2008
2012
2016
2020
2024

VMW:

ANN:


200
175
150
125
100
75
50
25
2007/12/26
2009/01/27
2010/03/01
2011/04/03
2012/05/05
2013/06/07
2014/07/10
2015/08/12
2016/09/13
2017/10/16
2018/11/18
2019/12/21
2021/01/22
2022/02/24
2023/03/29

SVM:


4
3
2
1
0
2008
2010
2012
2014
2016
2018
2020
2022
2024