МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«____»_____202___ р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЛАСИФІКАЦІЇ ПАРАМЕТРІВ ФІНАНСОВОГО РИНКУ

Спеціальність 122 «Комп'ютерні науки»

122 – КРМ – 601.21810528

Виконав студент 6-го курсу, групи 601
_____ О. В. .Шевченко
«19» лютого 2024 р.

Керівник: канд. техн. наук, доцент
_____Є. В. Сіденко
«19» лютого 2024 р.

Миколаїв – 2024

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____20___ р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
Шевченку Олександру Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи магістра «Інтелектуальна система класифікації параметрів фінансового ринку».
Керівник роботи Сіденко Євген Вікторович, канд. техн. наук, доцент.
Затв. наказом Ректора ЧНУ ім. Петра Могили від «01» лютого 2024 р. № 20
2. Строк подання студентом роботи 19 лютого 2024 р.
3. Вхідні (початкові) дані до роботи: новини поділенні за настроями на 3 категорії.
Очікуваний результат: система класифікації руху напрямку ціни активів.
4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):

- провести аналіз основних понять та визначень, пов'язаних із задачею класифікації росту активів;
- розглянути вплив аналізу настроїв на розвиток цін та роль ліквідності в інвестиційному процесі;
- дослідити останні дослідження та публікації в даній галузі;
- проаналізувати технології, такі як логістична регресія, KNN, SVM для вирішення задачі класифікації курсу та зробити їх порівняння;

– порівняти залежність позитивного і негативного сентименту на ціну активу, їх загальний вплив на ціну та значущість для вирішення цієї задачі.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: Захист від іонізуючих випромінювань.

7. Консультанти:

Розділ	**Прізвище, ініціали та посада консультанта**	**Підпис**
Спеціальна частина з охорони праці	д-р біол. наук., професор Григор'єва Л.І.	
Методична частина	канд техн. наук, доцент Сіденко Є.В.	

Керівник роботи __канд. техн. наук, доцент Сіденко Є.В._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання ___Шевченко О. В._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «__31__» _____жовтня____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи магістра

Тема: Інтелектуальна система класифікації параметрів фінансового ринку

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників КРМ	01.09.2023	10.10.2023	Виконано
2	Отримання завдання на виконання КРМ	19.10.2023	19.10.2023	Виконано
3	Складання календарного плану роботи на весь період виконання КРМ	11.11.2023	15.11.2023	Виконано
4	Отримання завдання на передатестаційну практику, огляд матеріалу для КРМ	16.11.2023	27.11.2023	Виконано
5	Проходження передатестаційної практики, збір та аналіз матеріалів до КРМ	27.11.2023	23.12.2023	Виконано
6	Розробка звіту з передатестаційної практики	19.12.2023	12.01.2024	Виконано
7	Опис фахової частини КРМ, аналіз сучасного стану задачі класифікації напрямку росту активів, огляд технологій для вирішення поставленої задачі, програмна реалізація	13.01.2024	25.01.2024	Виконано
8	Розробка спеціальної частини з охорони праці та надзвичайних ситуацій, методичної частини	26.01.2024	02.02.2024	Виконано
9	Перший попередній захист КРМ на засіданні комісії кафедри	29.01.2024	31.01.2024	Виконано
10	Другий попередній захист КРМ на засіданні комісії кафедри	12.02.2024	14.02.2024	Виконано
11	Доробка та остаточне оформлення КРМ	03.02.2024	11.02.2024	Виконано
12	Подання КРМ рецензенту	14.02.2024	16.02.2024	Виконано
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.02.2024	19.02.2024	Виконано
15	Захист КРМ перед ЕК	26.02.2024	27.02.2024	Виконано

Розробив студент _____Шевченко О. В._____ _____
(прізвище та ініціали) (*підпис*)

Керівник роботи ____канд. техн. наук, доцент Сіденко Є.В _ _____
(наук. ступінь, вчене звання, прізвище та ініціали) *(підпис)*

«____» _____ 202__р.

АНОТАЦІЯ

Кваліфікаційна робота магістра студента групи 601 ЧНУ ім. Петра Могили Шевченка Олександра Вікторовича

Тема: «Інтелектуальна система класифікації параметрів фінансового ринку»

Об'єкт дослідження – процес короткострокового прогнозування на фінансовому ринку.

Предмет дослідження – моделі регресійного аналізу для класифікації параметрів фінансового ринку.

Мета роботи – підвищення ефективності зростання ціни активу за рахунок застосування методів машинного навчання.

У першому розділі розглядається сучасний стан та публікації по темі роботи.

У другому розділі проведено аналіз регресійних моделей і обрано інструментальні засоби розробки застосунку.

У третьому розділі описано реалізацію застосунку, створення та використання моделей.

В результаті розроблено 3 моделі регресійного аналізу і порівняно їх якості.

Кваліфікаційна робота магістра містить 114 сторінок, 33 рисунків та 58 використаних джерел.

Ключові слова: *R, Регресивні моделі, класифікація курсу, крипто актив.*

ABSTRACT

Master's qualification work of a student of group 601 Petro Mohyla Black Sea National University

Shevchenko Oleksandr Viktorovych

Topic: "Intelligent system for classifying financial market parameters"

Object of research - the process of short-term forecasting in the financial market.

Subject of research - regression analysis models for classification of financial market parameters.

Purpose - to increase the efficiency of asset price growth by applying machine learning methods.

The first section reviews the current state of the art and publications on the topic.

The second section analyzes regression models and selects tools for developing the application.

The third section describes the application implementation, model creation, and use.

As a result, 3 models of regression analysis were developed and their qualities were compared.

Master's qualification work contains 114 pages, 33 figures and 58 references.

Keywords: R, Regression models, course classification, crypto asset.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

CRAN – Comprehensive R Archive Network

KNN – k-nearest neighbors algorithm

RSI – Relative Strength Index

SVM – Support Vector Machines

ВСТУП

Сьогоднішнє накопичення великих обсягів інформації вимагає компетентної обробки для прийняття обґрунтованих рішень. Проблеми математичного моделювання стають дедалі більш актуальними для ефективної організації управління різними суб'єктами господарювання та економічними спільнотами. Це відбувається через те, що якість прийнятих рішень в сучасному світі надзвичайно залежить від якості прогнозування можливих наслідків. Таким чином, сьогоднішні рішення повинні ґрунтуватися на достовірних оцінках можливого розвитку процесів та подій у майбутньому.

Головною метою є створення інтелектуальної системи, яка буде здатна проводити дослідження курсу активу і надавати прогноз щодо росту і падіння, що дасть змогу інвесторам зробити висновки щодо перспективності напрямків і доцільності інвестування.

В даній роботі всі дослідження і експерименти проводилися з набором реальних даних задля досягнення максимальної ефективності в задачі класифікації напрямку руху активу

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ КЛАСИФІКАЦІЇ НАПРЯМКУ РОСТУ АКТИВІВ

В цьому розділі наведені дослідження, на тему впливу настроїв на фондовий ринок і використання нейронних мереж для класифікації напрямку росту ціни. Також розглянуто дослідження що використовують, як контрольовані та неконтрольовані алгоритми машинного навчання, ансамблеві алгоритми, алгоритми аналізу часових рядів і алгоритми глибокого навчання для класифікації курсу акцій.

1.1 Основні поняття та визначення

Через непередбачувану природу фондового ринку окремим особам дуже важко отримати прибутки від своїх інвестицій. Первинний, фундаментальний і технічний аналіз є популярними підходами до розуміння ринкових тенденцій але вони мають властиві обмеження через залучення відставання показників і неточність прогнозу [1].

Це спонукало дослідників до розробки вдосконалених методів для ринкових сценаріїв у реальному часі на основі моделей машинного та глибокого навчання [2].

У сучасну епоху алгоритми машинного та глибокого навчання мають значні переваги перед традиційними методами, такими як технічний і фундаментальний аналіз. Використовуючи можливості машинного навчання та штучного інтелекту, ці алгоритми полегшують класифікації курсу цін на акції та індексів. Машинне навчання слугує додатковим підходом поряд із технічним і фундаментальним аналізом [2], поєднання цих інструментів утворює потужну торгову платформу.

Моделі машинного навчання можуть надавати рішення для таких проблем, як прогнозування та класифікація цін на акції, управління портфелем, алгоритмічна торгівля, аналіз настроїв на фондовому ринку, оцінка ризиків тощо. З цих проблем ця оглядова стаття зосереджена на дослідженні різних підходів, описаних для класифікації напрямку росту цін на акції. і класифікація.

Серед текстових даних, незважаючи на те, що було опубліковано багато робіт, які аналізують соціальні медіа [1], вмісту новин приділено мало уваги в класифікації руху фондового ринку [3]. Визнаючи, що новини мають вирішальне значення для класифікації напрямку фондового ринку, цей систематичний огляд зосереджений на документах, які досліджують машинне навчання та методи аналізу тексту для класифікації курсу фондового ринку за допомогою новин.

Іноді на рух ціни головним чином впливають настрої [2]. Ці настрої можуть бути позитивними, що призводять до бичачого руху, або негативними, що призводить до ведмежого руху.

Отже, аналіз настроїв акцій важливий для розуміння класифікації курсу цін на акції та класифікації трендів[4]. Крім того, у цьому розділі досліджується деякі методи класифікації курсу на основі аналізу настроїв. Дані соціальних мереж, новини компаній і аналіз тенденцій можуть класифікувати настрої інвесторів щодо акцій як позитивні, негативні чи нейтральні.

Коли текстові дані збираються, вони можуть містити правильні та фальшиві дані. В одному дослідженні [3], автори використовували вибір функцій, щоб усунути фейкові новини та спам-твіти, зібрані з даних соціальних мереж. Це покращило якість даних для навчання, і для навчання моделі використовувався алгоритм класифікації, алгоритм випадкового лісу.

Настрої можуть бути позитивними, нейтральними чи негативними, що допомагає людям вирішити, купувати чи продавати акції. Негативні настрої впливають на кон'юнктуру ринку [5].

Дослідження, у якому було проведено порівняння двох акцій, Tesla та Nio, на основі настроїв. Виявилося, що негативні події, такі як протест Tesla у 2021 році, вплинули і на її конкурента Nio. Це дослідження ґрунтувалося на історичних даних із використанням прогнозування часових рядів із даними за 10, 15 і 20 днів [3].

1.1.1 Аналіз настроїв як індикатор розвитку цін

Відповідно до концепції мікроекономіки, ціна акцій та інших цінних паперів є насамперед під впливом фундаментального закону попиту та пропозиції. Попит є Крім того, під впливом процентних ставок, корпоративних результатів та економічних даних. Це призводить до дуже мінливих і коливальних змін цін.

Кілька досліджень намагалися відповісти на питання, чи можна передбачити ціни на акції. Гіпотеза ефективного ринку Фама (1970) [5], стверджує, що вся доступна інформація негайно включається в ціни акцій.

Єдиним чинником для розвитку курсу акцій є нова інформація через те, що новини можна передбачити, ціни на акції також не можуть бути передбачені [3].

Хоча пізніші дослідження можуть довести, що деякі припущення EMH нереалістичні, а також те, що прогнози до певної точки можуть бути можливими. Інші теорії, такі як теорія хвиль Елліота [6], підтверджують існування рекурентних моделей що слідують ціни на акції. Окрім технічного аналізу, це свідчать емпіричні дослідження деякі імпульсні стратегії мають прибуткову прогностичну силу.

Крім того, згідно з різними дослідженнями, описаними нижче, зміни цін на акції корелюють дані про настрої в соціальних мережах. Завдяки збільшенню обчислювальної потужності за останні роки машинне навчання стало здатний обробляти й оцінювати величезні обсяги даних.

Шаблони та відносини, які важко розпізнати за допомогою математичних і статистичних підходів, можна помітити з цими останніми техніками [7].

1.1.2 Вплив ліквідності та настроїв інвесторів на стадну поведінку

Tingyu Zhou і Lai [8] зазначають, що стадна поведінка особливо популярна для невеликих акцій та під час економічного спаду, і що інвестори, швидше за все, будуть продавати цілими стадами, ніж купувати акції. Вони стверджують, що стадна поведінка може бути короткочасною і виникає лише в певній галузі, припускаючи, що стадна поведінка не є постійним явищем. Хоча стадна поведінка широко досліджується на фінансових ринках [9-13] його вивчення на ринку криптовалют все ще обмежене.

По-перше, криптовалюти пережили швидкий розвиток і стали популярними активами на світових фінансових ринках, привертаючи увагу інвесторів і політиків головним чином завдяки своїм унікальним характеристикам. Крім того, ринок криптовалют має такі відмінні характеристики, як висока волатильність, великий розмір і значна неоднорідність [7].

По-друге, моделі поведінки, які спостерігаються серед криптоінвесторів, свідчать про те, що вони схильні брати участь у торгівлі, керованій настроями та обсягом, часто зосереджуючись на короткочасних трендах. Така поведінка, яка характеризується використанням погодинної та щоденної частоти для угод зі значним настроєм та обсягом, підтверджує поширеність галасливої торгівлі на ринку криптовалют.

По-третє, незважаючи на зростаючу важливість інституційних інвесторів на ринку криптовалюти, оскільки сектор отримує визнання серед широкої громадськості, значну частину ринку все ще займають індивідуальні інвестори, які часто менш поінформовані та менш обережні порівняно з інституційні інвестори.

Ґрунтуючись на наукових досягненнях, постулюється, що на феномен стадної поведінки на ринку криптовалют можуть впливати як ліквідність, так і настрої. Крім того, нерегульований характер цього ринку та переважання окремих інвесторів із порівняно нижчим рівнем знань ще більше посилюють вищезазначений вплив [8]. Одним із відповідних компонентів неліквідності ринку є інформаційна асиметрія між «інформованими» та «неінформованими» трейдерами.

Оскільки фундаментальні цінності на ринку криптовалюти не відчутні, інвестрами можуть керувати проінформовані трейдери, які діють на основі своїх особистих сигналів, особливо коли проінформовані трейдери отримують узгоджені сигнали, які стосуються купівлі чи продажу.

Отже, вважається, що настрої та інтенсивність ліквідності мають відчутний вплив на ринок криптовалюти, що вимагає дослідження їхнього впливу на поведінку стада в межах трьох різних категорій криптовалюти, що характеризуються різними розмірами.

Boxiang Jia стверджує що настрої інвесторів можуть бути потенційно вирішальним фактором, що стимулює стадну поведінку[8]. І виступає за нові варіанти настроїв інвесторів на ринку криптовалют і визначає надзвичайно позитивні (негативні) настрої як ейфорію (дисфорію). Що під час ейфорії та дисфорії спостерігається значна стадність.

Стадність - це поведінка, коли люди копіюють поведінку інших людей, навіть якщо вона є нелогічною або шкідливою. В контексті інвестування стадність може призвести до спекуляцій або паніки на ринку.

На рисунку 1.1 зображена діаграма що показує взаємозв'язок між настроєм інвестора, дисфорією та стадністю.

Також вирізняється три типи настрою:

Ейфорія - це стан надмірного оптимізму та ентузіазму. Інвестори в стані ейфорії часто приймають необдумані рішення, які можуть призвести до збитків.

Дисфорія - це стан надмірного песимізму та тривоги. Інвестори в стані дисфорії часто уникають інвестування, що може призвести до упущеної вигоди.

Нейтральний настрій - це стан, коли інвестор не відчуває ні ейфорії, ні дисфорії. Інвестори в нейтральному настрої приймають більш обґрунтовані рішення.

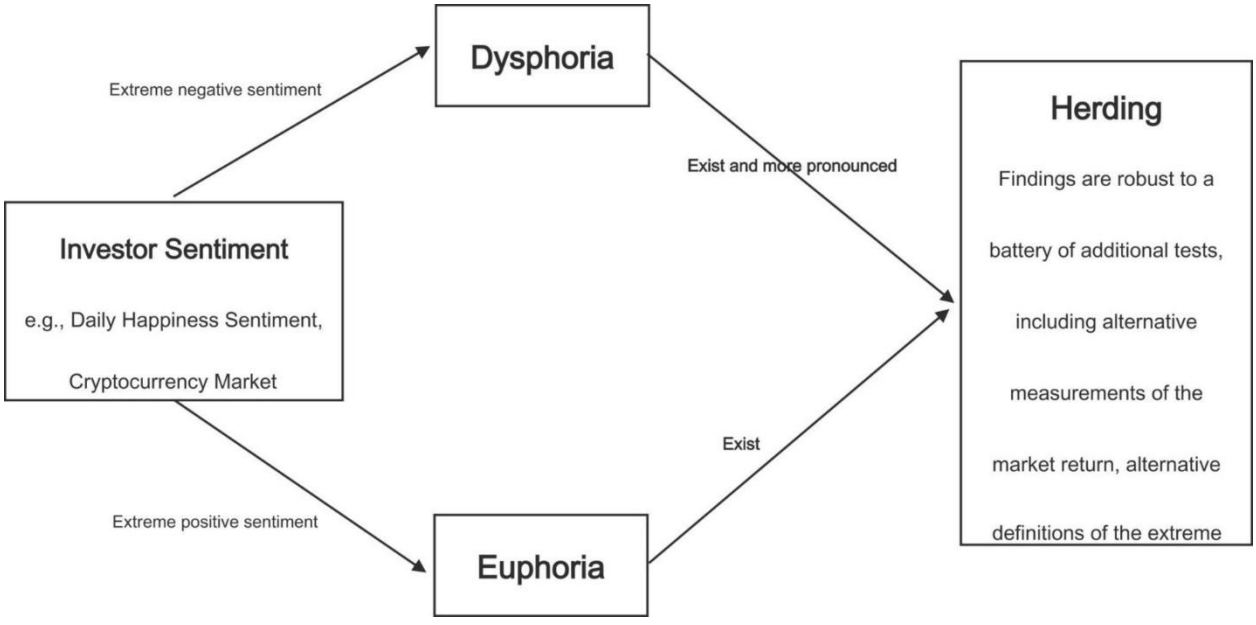

Рисунок 1.1 – Взаємозв'язок між настроєм, дисфорією та стадністю [8]

Нижче наведено основні визначення з рисунку 1.1:

– “Investor Sentiment” - це загальний настрій інвесторів щодо фінансового ринку;

– “Dysphoria” - це стан надмірного песимізму та тривоги;

– “Herding” - це поведінка, коли люди копіюють поведінку інших людей, навіть якщо вона є нелогічною або шкідливою;

– “Findings are robust” - це означає, що результати дослідження були повторені в різних дослідженнях;

– “Battery of additional tests” - це означає, що дослідження було проведено за допомогою набору додаткових тестів;

– “Daly Happiness Sentiment” - це вимірювання загального щастя населення на основі опитувань;

– “Cryptocurrency Market return” - це прибуток або збиток на криптовалютному ринку;

– “Alternative definitions of the extreme” – альтернативні визначення екстремальних станів настрою.

1.2 Останні дослідження та публікації

Область аналізу настроїв та її використання в аналізі фондового ринку набуває все більшого інтересу, і те саме стосується літературного покриття. Перші дослідження щодо впливу інвесторських настроїв на майбутнє компанії були проведені в 2006 році. З заснуванням Twitter вперше і, одночасно, найвідоміше дослідження щодо впливу настроїв у Twitter на ціни акцій було опубліковано Йоханом Болленом і Хуніа Мао в 2011 році [8].

Боллен і Мао досліджували вплив твітів на індекс Доу-Джонса. Вони змогли довести кореляцію між настроєм в Twitter та розвитком індексу Доу-Джонса через 3-4 дні. Для аналізу настрою використовувалися різні інструменти:

Opinion Finder, класичний інструмент, що класифікує полярність тексту як позитивну або негативну, не показав жодної кореляції.

Google Profile of Mood States, з іншого боку, показав обіцяючі результати. Він розподіляє ставлення тексту за шість вимірів: “Sure”, “Kind”, “Alert”, “Happy”, “Vital”, and "Calm” показав точність 87,6% для передбачення тенденції цін Доу-Джонса [9]. Проте варто зазначити, що для оцінки цього показника використовувалось лише 19 днів, що може бути сприйнято як недостатність глибини тестової фази: "З 1 по 19 грудня 2008 року було обрано тестовий період через стабілізацію значень DJIA після значної волатильності в попередні місяці та відсутність надзвичайних або значущих соціокультурних подій" [9].

Шпренгер [10] зосередилися виключно на твітах, пов'язаних з фондовим ринком, для прогнозування трьох класів: "Купуй", "Тримай" і "Продавай". Мультимножинна наївна модель Байєса була навчена з використанням 2,500 твітів. На початковому етапі не було досягнуто миттєвого успіху, оскільки значущої кореляції не вдалося довести. Проте вони визначили, що користувачі з великою кількістю фоловерів та твіти з великою кількістю ретвітів призводили до кращих результатів прогнозування.

Хуанг та ін. [11] поєднали класифікатори настроїв з графіками свічок для передбачення рухів цін на фондовому ринку. Вони довели, що комбінація обох методів є більш ефективною, ніж їх використання окремо. Для передбачення вони використовували п'ять різних акцій - Apple, Tesla, IBM, Amazon, Alphabet. Для фільтрації непотрібної інформації вони включали тільки твіти з мінімум 5 ретвітами як індикатори настрою. Для аналізу настроїв використовувався лексиконний підхід з використанням VADER. Для оцінки прогнозного горизонту в 4-10 днів досягнута найвища точність при передбаченні тенденції цін протягом 10 днів. Звісно, точність варіювалася для різних акцій, з точністю 75.38% для Apple і 67.34% для Google.

Дослідження Курова та ін. [12] вивчало агреговані дані настроїв від Bloomberg, отримані з твітів, пов'язаних з акціями. В дослідженні було прийнято висновок, що, крім тенденцій цін на акції, настрій в Twitter містить інформацію щодо змін рекомендацій аналітиків, змін цільових цін, квартальних сюрпризів у

заробітку та відкриття цін на IPO". Крім того, було зазначено, що вплив настроїв в Twitter є ще значущішим для акцій з меншим аналітичним охопленням.

Користуючись двома різними підходами, Хандлозер [13] намагався передбачити розвиток цін акцій, що входять до складу індексів DAX і Dow Jones, використовуючи настрій в Twitter. Перший підхід передбачав використання настрою як індикатора, другий — це "пряме передбачення" [14]. Щодо прямого передбачення з використанням настрою як індикатора, оброблені твіти подавалися безпосередньо в рекурентну нейронну мережу. Хоча існували значні відмінності в точності передбачення між різними акціями, точність підходу аналізу настроїв досягала до 49%, а точність прямого передбачення досягала до 61% [15].

Непрямим підходом до вимірювання настроїв інвесторів, який широко застосовується, є побудова індексу настроїв на основі кількох проксі. Бейкер і Вурглер [16] створили зведений індекс настроїв із шести наближених: дисконт закритого фонду, ринковий оборот, кількість первинних публічних пропозицій (IPO) і прибутковість першого дня торгівлі акціями на IPO, новий випуск акцій і дивідендної премії.

Бейкер та ін. вилучили 3 змінні, включаючи дисконт закритого фонду, нову емісію акцій і дивідендну премію, з набору проксі-компонентів, наданих Бейкером і Вурглером, і додали премію за волатильність як нову проксі для індексу настроїв.

У роботі інших дослідників [17-20] настрої інвесторів також розпізнаються шляхом інтерпретації деяких торговельних дій, включаючи запозичення маржі, зміну короткострокових процентів і короткі продажі спеціалістів [17]. Індексу настроїв інвесторів вдається представити настрої ринку за певний період, але йому не вдається виміряти, як швидко інвестори реагують на нову інформацію на ринку.

Згідно з Li et al. [16], зростає інтерес до текстового аналізу настроїв серед дослідників фінансової поведінки, оскільки цей метод може зменшити упередженість, яка може бути виявлена в підході до настроїв на основі опитування використовували настрої новин, пов’язані з кожною фірмою, щоб стати проксі настроїв інвесторів щодо акцій цієї фірми. «Загальна оцінка настрою» в діапазоні від −1 до 1 відображає рівень оптимізму чи песимізму.

Чим вищий показник, тим більший оптимізм інвестора щодо акцій. Балі та ін. (2016) [21] заявив, що підвищення волатильності ринку пов’язане з незвичайними новинами. Незвичайні новини викликають розбіжності інвесторів щодо оцінки фірм. «Враховуючи високу вартість коротких продажів, песимістичні інвестори сидять осторонь, тоді як оптимістичні інвестори підвищують ціни на акції, щоб відобразити їхню власну оцінку» [22].

Ґрунтуючись на наведених дослідженнях [12-27], рішення про класифікацію напрямку фондового ринку отримано на основі фундаментального та технічного аналізу з аналізом настроїв і багатьма моделями навчання. Алгоритми, які використовуються для класифікації курсу фондового ринку на основі дослідницьких статей, наведено в таблиці 1.1

Таблиця 1.1 – Використання алгоритмів у прогнозуванні цін активів

Тип алгоритму	Короткий опис
LSTM, RNN, Logistic Regression, Naïve Bayes	Моделі прогнозування за формулами в 10 технічних індикаторів [5]
XG-Boost і LSTM, Random Forest	Класифікація курсу двох акцій – Tainwala Chemicals and Plastics (Mumbai, India) Lt. (TAINIWALCHM); – Agro Phos (Indore, India) Ltd. (AGROPHOS) [6].
NLP	Створення композитного індексу настроїв з шести проксі [8]: – дисконту закритого фонду; – ринкового обороту; – кількості первинних розміщень і публічних; – прибутковості першого дня торгів акціями; – акцій нової емісії та дивідендної премії.
CSAD і CSSD	Вибірка зі 100 криптовалют, щоб визначити, чи присутня стадна поведінка на ринку в цілому [13]
Baseline Regressions	Визначення ціни активу за середні значення – дивідендної премії – частки акцій у нових випусках – середня прибутковість за перший день[15]

1.3 Постановка задачі

Актуальність дослідження: у зв'язку зі зростаючим інтересом до аналізу настроїв в інвестиційному середовищі, важливість прогнозування ринкових тенденцій та розвитку ефективних методів в даній області надзвичайно актуальна для фінансових аналітиків та інвесторів.

Об’єкт дослідження – процес короткострокового прогнозування на фінансовому ринку.

Предмет дослідження – моделі регресійного аналізу для класифікації параметрів фінансового ринку.

Мета роботи – підвищення ефективності зростання ціни активу за рахунок застосування методів машинного навчання.

Таким чином, для досягнення поставленої мети необхідно виконати наступні завдання:

1) провести аналіз основних понять та визначень, пов'язаних із задачею класифікації росту активів;

2) розглянути вплив аналізу настроїв на розвиток цін та роль ліквідності в інвестиційному процесі;

3) дослідити останні дослідження та публікації в даній галузі;

4) проаналізувати технології, такі як логістична регресія, KNN, SVM для вирішення задачі класифікації курсу та зробити їх порівняння;

5) порівняти залежність позитивного і негативного сентименту на ціну активу, їх загальний вплив на ціну та значущість для вирішення цієї задачі.

Висновки до розділу 1

У розділі, присвяченому дослідженню впливу настроїв на фондовий ринок, розглянуто різні аспекти цієї проблематики. Початковій частині присвячено визначенню основних термінів та понять, пов'язаних з цією тематикою. Зазначено, що через непередбачувану природу фондового ринку отримати прибуток від інвестицій стає вельми складно для окремих осіб.

Розглянуті аспекти аналізу настроїв як індикатора розвитку цін. Вказано, що відповідно до концепції мікроекономіки, ціна акцій є результатом взаємодії закону попиту та пропозиції, а також інших факторів, таких як процентні ставки та корпоративні результати. Підкреслено важливість аналізу настроїв як індикатора ринкових тенденцій, зокрема на ринку криптовалют, де спостерігається висока волатильність та стадна поведінка інвесторів.

Детально проаналізовано статтю Boxiang Jia, який стверджує, що настрої інвесторів можуть впливати на стадну поведінку, що зазвичай призводить до спекуляцій та паніки на ринку. Вказано, що ринок криптовалют має свої особливості, зокрема швидкий розвиток та високу волатильність, що робить його особливо чутливим до впливу настроїв інвесторів.

У підсумку, зазначено необхідність подальших досліджень у цій галузі, а також використання передових технологій, таких як логістична регресія та інші методи машинного навчання, для аналізу та прогнозування руху цін на активи на фондовому ринку.

2 ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Огляд моделей для класифікації

Виходячи з досліджень для класифікації цін активів використовується велика кількість алгоритмів та моделей. Основні з яких можна поділити на 4 категорії. Загальну структуру можна переглянути на рисунку 1.2

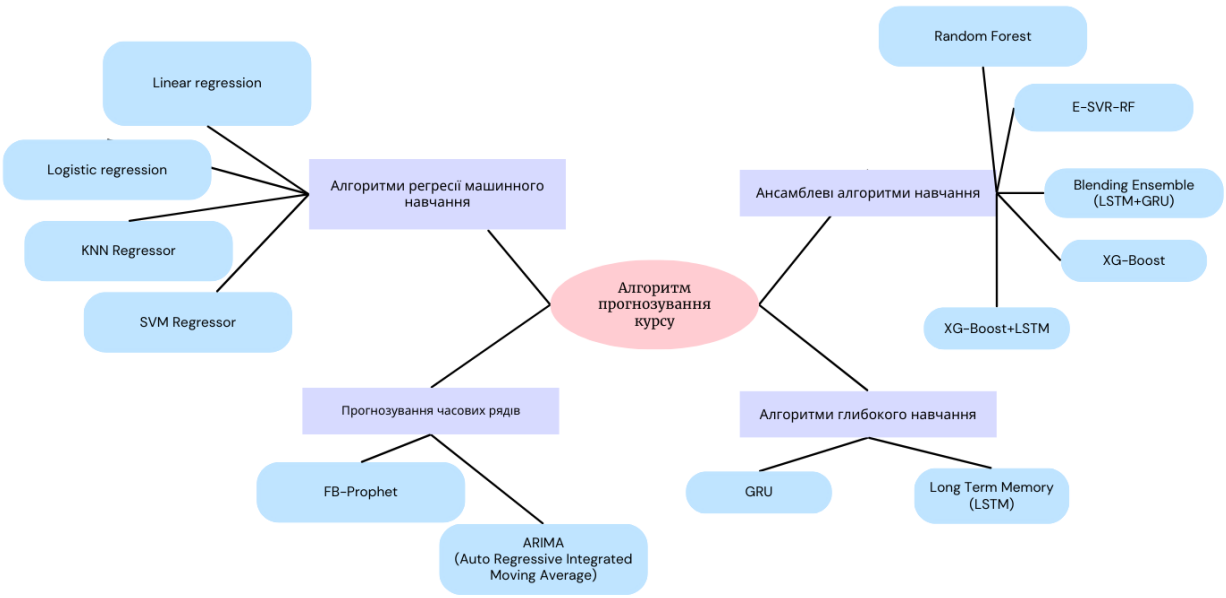

Рисунок 2.1 – Алгоритми класифікації активів

У рамках роботи було обрано 3 моделі для реалізацій і потім подальшого порівняння. А саме:

– логістична регресія;
– KNN;
– SVM.

2.1.1 Логістична регресія

Логістична регресія – це статистичний регресійний метод, який використовується для задач бінарної класифікації Використовуючи змінні для логістичних кривих, логістична регресія групує кілька незалежних факторів у дві або більше взаємовиключних груп і прогнозує ймовірність успішних акцій [28]. Найбільша перевага полягає в тому, що метод можна використовувати як для

класифікації, так і для оцінки ймовірності класу, оскільки він пов’язаний із розподілом логістичних даних. Метод бере лінійну комбінацію ознак і застосовує до них нелінійну сигмоподібну функцію .

У базовій версії логістичної регресії вихідна змінна є двійковою, однак її можна розширити до кількох класів (тоді це називається мультиноміальною логістичною регресією) [29]. Бінарна логістична модель класифікує зразки на два класи, тоді як багатономіальна логістична модель розширює це до довільної кількості класів без їх упорядкування.

Математика логістичної регресії спирається на концепцію «шансів» події, яка є ймовірністю події, поділеною на ймовірність того, що подія не відбудеться. Як і лінійна регресія, логістична регресія має ваги, пов’язані з розмірами вхідних даних. На відміну від лінійної регресії, зв’язок між вагами та виходом моделі є експоненціальним, а не лінійним [28].

Логістична регресія є популярним методом в машинному навчанні і статистиці для вирішення задач класифікації, особливо у випадках, коли залежна змінна є дихотомічною (тобто приймає два можливі значення, наприклад, "так" або "ні"). Вона дозволяє оцінити ймовірність того, що дана подія відбудеться, на основі однієї або декількох предикторних змінних.

Логістична регресія базується на логістичній функції, також відомій як сигмоїда, яка приймає будь-яке дійсне число і перетворює його у значення між 0 і 1, що інтерпретується як ймовірність.

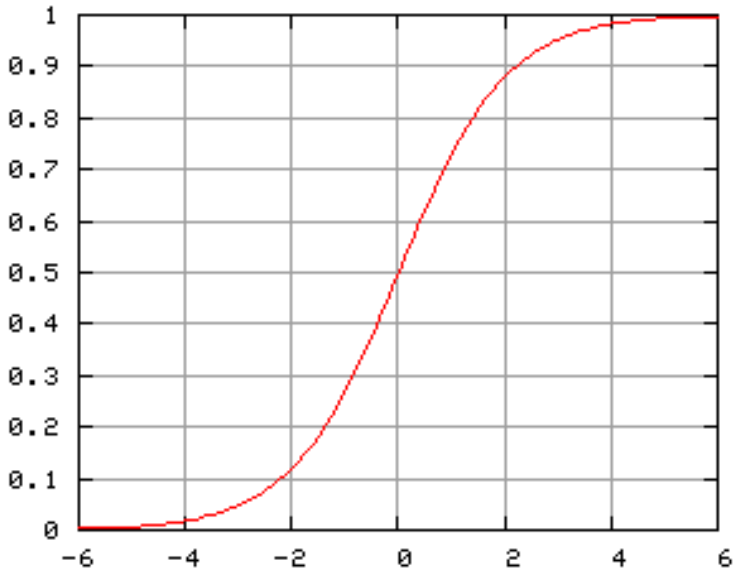

Рисунок 2.2 – Приклад Сигмоїди [29]

Формула логістичної регресії виглядає наступним чином[28]:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)}} \quad (2.1)$$

де $P(Y = 1)$ є ймовірністю того, що залежна змінна Y приймає значення 1;

$X1$, $X2$, ..., Xk є незалежними змінними;

$\beta 0$, $\beta 1$, ..., βk є параметрами моделі, що оцінюються за допомогою методу максимальної правдоподібності.

Логістична регресія широко застосовується в різних областях, включаючи медицину для прогнозування ризику захворювань, фінанси для оцінки кредитного ризику, маркетинг для прогнозування відгуку клієнта на рекламні кампанії, та соціологію для аналізу соціальних тенденцій.

Основною перевагою логістичної регресії є її здатність працювати з дихотомічними залежними змінними та інтерпретувати результати як ймовірності. Вона також дозволяє включати як кількісні[28], так і категорійні незалежні змінні, роблячи її гнучким інструментом для аналізу даних.

Однак, серед обмежень логістичної регресії можна відзначити припущення про лінійність відносин між логарифмом шансів і незалежними змінними, а також чутливість до високої кореляції між предикторами. Також, вона може не ефективно

працювати з дуже складними взаємозв'язками або великою кількістю категорійних змінних.

Приклади використання:

1) **медичний аналіз ризику захворювання**. У медицині логістична регресія може бути використана для прогнозування ймовірності розвитку певного захворювання на основі клінічних і демографічних даних пацієнта. Наприклад, модель може оцінювати ризик серцево-судинних захворювань враховуючи такі фактори, як вік, стать, кров'яний тиск, рівень холестерину, куріння та історію сімейних захворювань. Кожен із цих факторів вносить свій вклад у загальний ризик, який потім може бути представлений у формі ймовірності;

2) **оцінка кредитного ризику.** В області фінансів логістична регресія може бути застосована для оцінки ймовірності невиконання кредитних зобов'язань клієнтом. Модель може використовувати різноманітні фінансові індикатори, такі як кредитний рейтинг, історія платежів, рівень доходу та зайнятість, для визначення ймовірності дефолту. Це дозволяє банкам та іншим кредитним організаціям ефективно управляти ризиками;

3) **прогнозування відгуку на маркетингові кампанії.** У маркетингу логістична регресія може бути використана для прогнозування ймовірності, з якою потенційний клієнт відреагує на певну рекламну кампанію, наприклад, відправляючи заявку або здійснюючи покупку. Модель може аналізувати дані про попередні взаємодії клієнта з брендом, демографічні характеристики, інтереси та поведінку в соціальних мережах для визначення найбільш зацікавлених і потенційно реагуючих сегментів аудиторії;

4) **соціологічні дослідження.** У соціології логістична регресія може бути використана для аналізу впливу різних соціальних і економічних факторів на певні поведінкові вибори або думки людей. Наприклад, дослідження може досліджувати, як вік, освіта, дохід та місце проживання впливають на політичні переконання або готовність участі у громадських акціях[29].

2.1.2 K-найближчих сусідів (KNN)

KNN — це метод класифікації та регресії, який називають ледачим учнем, оскільки для навчання не потрібен великий період часу. Однією з переваг KNN є те, що це один із найпростіших алгоритмів ML.

Єдина дія, яку потрібно виконати для KNN, це обчислити значення K і евклідову відстань [30]. Аспект повільного навчання цього алгоритму робить його швидшим, ніж інші алгоритми. Він може погано узагальнюватися для великих даних, оскільки він пропускає етап навчання. Розрахунок евклідової відстані наведено у рівнянні (2.2).

$$D(h_i, p_r) = \sqrt{\sum_{l=1}^{n} (P_r - h_i)^2} \quad (2.2)$$

Метод k-найближчих сусідів (KNN) вважається одним з найпростіших, але при цьому дуже ефективних алгоритмів машинного навчання, що використовуються для класифікації та регресії.

Він базується на простій інтуїції: для прогнозування значення залежної змінної для нового спостереження, алгоритм ідентифікує *k* найближчих до цього спостереження точок у навчальному датасеті і використовує їхні значення для визначення результату.

Ці "найближчі" точки визначаються за допомогою метрик відстані, таких як евклідова, Манхеттенська або Мінковського, що дозволяє кількісно оцінити схожість між спостереженнями [31].

Вибір оптимального значення k у методі k-найближчих сусідів (KNN) є критично важливим з кількох причин, що безпосередньо впливають на ефективність та точність моделі. Ось чому підбір k є важливим:

1) **зменшення шуму та уникнення перенавчання**. Коли k занадто мале, модель стає високочутливою до шуму у навчальних даних. Це означає, що навіть незначні варіації в даних можуть сильно вплинути на результат класифікації, ведучи до перенавчання (overfitting), коли модель добре працює на навчальному наборі даних, але погано — на нових, не бачених раніше даних;

2) **загальність моделі та уникнення недонавчання**. Навпаки, дуже велике значення k може призвести до того, що модель буде занадто загальною, не звертаючи увагу на більш тонкі відмінності між класами. У цьому випадку модель може не розпізнавати важливі патерни в даних, що веде до недонавчання (underfitting), коли модель не може адекватно відобразити структуру навчального набору даних і, як наслідок, має низьку точність на як навчальному, так і на тестовому наборах даних;

3) **баланс між згладжуванням межі рішення та адаптацією до даних.** Оптимальне значення k допомагає знайти правильний баланс між здатністю моделі адаптуватися до складності даних і потребою уникнути зайвої чутливості до окремих точок даних. Це дозволяє формувати більш гладкі та стабільні межі рішення, які краще узагальнюються на нових даних;

4) **вплив на обчислювальну складність**. Значення k також впливає на обчислювальну складність моделі. Більші значення k вимагають більше часу для визначення найближчих сусідів і, відповідно, для класифікації нових точок даних. Отже, оптимальне k також має враховувати обмеження на ресурси та вимоги до швидкодії моделі;

5) **варіативність даних**. Різні набори даних можуть мати різну структуру та рівень складності, тому k, яке працює добре для одного набору даних, може бути неідеальним для іншого. Експериментування з різними значеннями k дозволяє знайти найкращий варіант для конкретного набору даних.

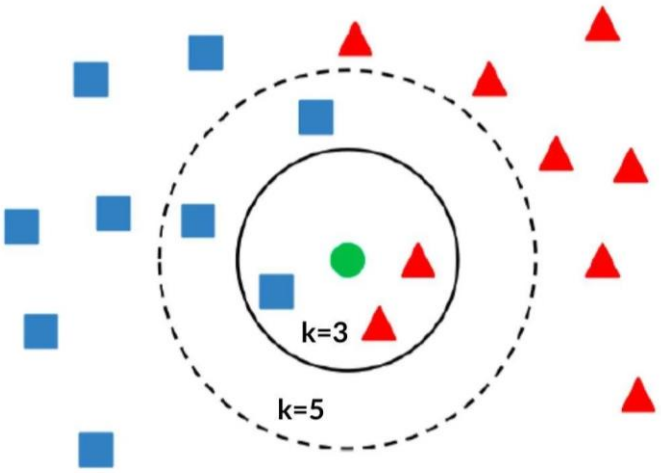

Рисунок 2.3 –Візуалізація моделі K-найближчих сусідів [32]

У контексті класифікації, KNN призначає новому спостереженню клас, який є найбільш поширеним серед його *k* найближчих сусідів. У випадку регресії, вихідне значення для нового спостереження обчислюється як середнє (або інша форма агрегації) значень *k* найближчих сусідів. Один з ключових аспектів ефективності KNN полягає у виборі параметра *k*, який впливає на здатність моделі уникати перенавчання при надто малому значенні або втрати чутливості до локальних особливостей даних при надто великому *k*.

KNN відомий своєю простотою у використанні та високою адаптивністю до конкретних задач. Він не вимагає складних математичних обчислень для тренування моделі, оскільки "навчання" полягає лише у зберіганні навчального датасету. Проте, це також означає, що KNN може бути відносно вимогливим до ресурсів при великих обсягах даних через необхідність обчислення відстаней між новим спостереженням та кожною точкою в датасеті [33].

Однак, KNN продемонстрував свою ефективність у широкому спектрі застосувань, від рекомендаційних систем і пошуку схожих документів до біомедичних досліджень і фінансового аналізу. Ефективність KNN може бути значно покращена за допомогою технік зменшення розмірності даних, таких як головні компоненти аналіз або вибіркове відкидання ознак, які дозволяють зменшити обчислювальне навантаження і покращити точність прогнозування.

Незважаючи на свою простоту, KNN залишається важливим інструментом у наборі засобів дослідника, завдяки своїй здатності до гнучкого застосування в різноманітних ситуаціях і легкості в інтерпретації результатів. Його використання як базової лінії для порівняння з більш складними моделями часто дозволяє отримати цінні інсайти про структуру даних і потенціал для подальшого покращення прогнозних моделей [32].

Нижче наведено кілька прикладів використання KNN у різних сферах[34]:

1) **рекомендаційні системи**. KNN широко використовується для розробки рекомендаційних систем, наприклад, у сервісах стрімінгу музики або фільмів. Алгоритм аналізує схожість між користувацькими профілями або об'єктами (фільми, пісні) на основі історії переглядів або прослуховувань, а потім рекомендує користувачеві контент, популярний серед сусідів з подібними смаками;

2) **класифікація зображень**. У комп'ютерному зорі KNN може бути використаний для класифікації зображень, ідентифікації об'єктів або розпізнавання облич. Алгоритм порівнює нові зображення з відомими зразками у базі даних і класифікує їх на основі схожості з найближчими сусідами;

3) **медична діагностика**. У медицині KNN використовується для допомоги в діагностиці захворювань, наприклад, у класифікації ракових пухлин як злоякісних або доброякісних на основі даних медичних знімків або лабораторних аналізів. Метод дозволяє враховувати схожі випадки з історії та з великою точністю визначати тип пухлини;

4) **фінансовий аналіз**. У фінансовому секторі KNN може бути застосований для оцінки кредитоспроможності клієнтів або для прогнозування банкрутства компаній. Аналізуючи фінансові показники та порівнюючи їх із історичними даними про успішні та неуспішні кредитні випадки, алгоритм допомагає визначити ризики;

5) **виявлення шахрайства**. KNN також використовується в системах виявлення шахрайства, наприклад, при аналізі транзакцій кредитних карт. Система може ідентифікувати потенційно шахрайські транзакції, порівнюючи їх із зразками нормальної поведінки користувачів та відомими шахрайськими схемами.

2.1.3 Метод опорних векторів (SVM)

Метод опорних векторів передбачає контрольоване навчання, яке використовується для категоризації аспектів за допомогою розділювача. Потім роздільник виявляється, коли дані спочатку відображаються у просторі ознак великої розмірності. Він знаходить категоризацію точок даних у n-вимірному просторі та знаходить оптимальну гіперплощину.

Точки даних групуються відповідно до їхнього розташування відносно гіперплощини.

Продуктивність алгоритму SVM можна підвищити, налаштувавши такі параметри, як регулярізація, гама та параметри ядра. SVM також можна використовувати для аналізу настроїв, щоб оцінити настрої інвесторів, які опосередковано вплинуть на ринкові умови. Він добре підходить як для великорозмірних наборів даних, так і для невеликих наборів даних [35].

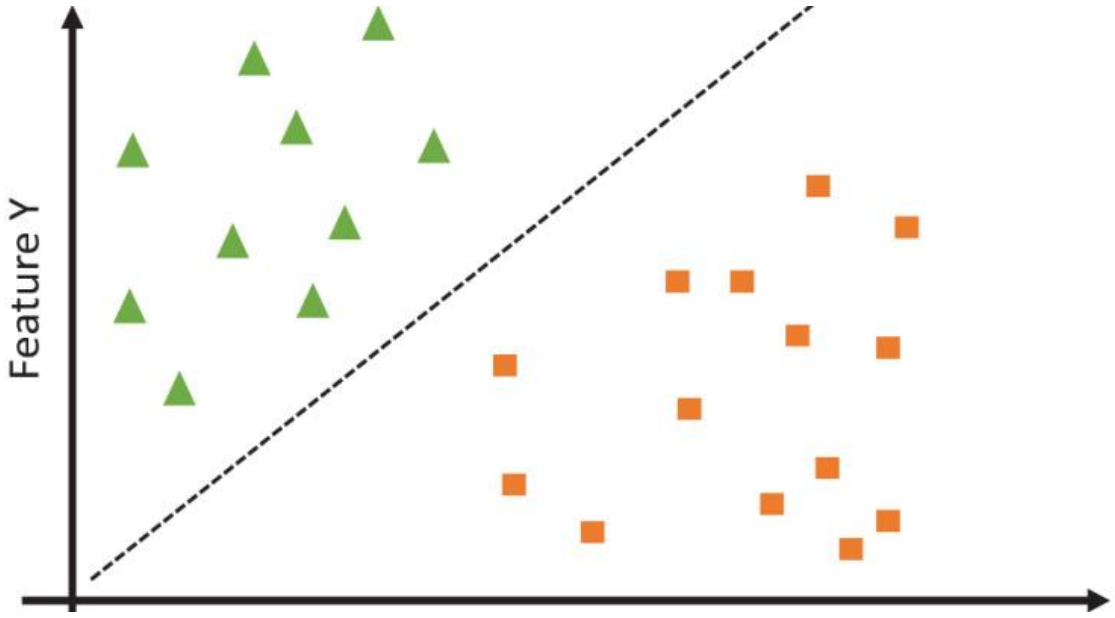

Рисунок 2.4 – Ілюстрація лінійного SVM [36]

Основна ідея SVM полягає в пошуку гіперплощини в просторі ознак, яка найкраще розділяє дані на класи.

У випадку класифікації, SVM намагається максимізувати відстань між найближчими точками різних класів, які називаються опорними векторами, що дозволяє досягти найкращого розділення.

Однією з ключових особливостей SVM є використання ядерних функцій, що дозволяє ефективно працювати в просторах високої розмірності, навіть якщо вихідні дані не є лінійно роздільними в оригінальному просторі ознак. Це означає, що за допомогою ядерного трюку SVM може знаходити складні нелінійні межі рішень, не підвищуючи обчислювальну складність моделі [37].

SVM широко застосовується у багатьох сферах. Його ефективність у складних задачах класифікації, зокрема, робить його популярним вибором серед науковців і інженерів.

Проте, ефективність SVM сильно залежить від вибору ядра та його параметрів, а також від параметра регуляризації, який контролює баланс між максимізацією відстані маржі та мінімізацією помилки класифікації.

Правильний вибір цих параметрів може значно покращити якість моделі, але водночас вимагає додаткових зусиль у вигляді підбору параметрів та валідації [37].

Незважаючи на ці виклики, SVM залишається одним із найефективніших інструментів для розв'язання багатьох складних задач аналізу даних. Його здатність виявляти складні шаблони в даних із забезпеченням високої точності прогнозування робить його незамінним інструментом.

Формула для методу опорних векторів (SVM) в контексті лінійної класифікації визначає гіперплощину, яка розділяє два класи. Гіперплощина описується рівнянням:

$$w \cdot x - b = 0 \tag{2.3}$$

де:

- w є вектором ваг, що орієнтує гіперплощину;
- x є вектором ознак вхідного прикладу;
- b є зсувом гіперплощини від початку координат.

У випадку нелінійної класифікації, SVM використовує ядерні функції для перетворення вхідних даних в простір вищої розмірності, де лінійне розділення стає можливим. В цьому контексті, функція рішення використовує ядерну функцію

$K(x, x_i)$ для обчислення скалярного добутку між векторами у перетвореному просторі [38]:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \, K(x, x_i) - b \qquad (2.4)$$

- N є кількістю опорних векторів;
- α_i є коефіцієнтами, які визначаються під час тренування моделі;
- уі є мітками класів опорних векторів;
- $K(x, x_i)$ є ядерною функцією, що вимірює схожість між вхідним вектором x та опорним вектором x_i.

Приклади використання SVM:

1) **розпізнавання образів**. SVM широко використовується у комп'ютерному зорі для розпізнавання облич, рукописних цифр, та інших об'єктів на зображеннях. Завдяки здатності обробляти велику кількість ознак та ефективно класифікувати об'єкти, SVM демонструє високу точність у цих задачах;

2) **класифікація текстів.** У обробці природних мов SVM використовується для класифікації текстів, наприклад, для визначення тональності відгуків (позитивний, негативний), класифікації електронних листів на спам та не спам, або ж для автоматичного розподілу статей за категоріями;

3) **біоінформатика**. У біоінформатиці SVM застосовують для класифікації біологічних даних, наприклад, для прогнозування функцій білків, класифікації ракових тканин на основі геномних даних або для розпізнавання патогенних організмів;

4) **фінансовий аналіз**. В економіці та фінансах SVM використовують для прогнозування банкрутства компаній, аналізу ринкових тенденцій, оцінки ризику кредитування та інших задач прогнозування, де потрібна висока точність та надійність;

5) **розпізнавання голосу**. SVM також застосовується в системах розпізнавання голосу для класифікації аудіо сигналів та визначення мовця або

команд, що значно покращує інтерактивність та функціональність голосових інтерфейсів [35].

2.2 Мова програмування R

R — це інструмент статистичного програмування, який унікально обладнаний для обробки даних і багатьох із них.

З R легко працювати над величезними обсягами інформації та створювати готові до публікації графіки та візуалізації. Так само, як і всілякі завдання аналізу даних, аналізу даних і моделювання.

Оскільки вперше він був розроблений статистиками для статистичних цілей, R надзвичайно добре підходить для науки про дані, важливої галузі в сучасному світі.

Хоча основною функцією R є статистичний аналіз і графіка, його використання поширюється не тільки на штучний інтелект, машинне навчання, фінансовий аналіз тощо. Визнана однією з найпопулярніших мов програмування у світі. R існує з початку.

1990-х років і досі розвивається. R – система статистичних обчислень і графіка. Ця система складається з двох частин: самої мови R (саме це те, що більшість людей мають на увазі, коли говорять про R) і середовища виконання.

R є інтерпретованою мовою, що означає, що користувачі отримують доступ до її функцій через інтерпретатор командного рядка [39].

На відміну від таких мов, як Python і Java, R не є мовою програмування загального призначення. Натомість він вважається доменно-орієнтованою мовою (DSL), тобто його функції та використання призначені для певної сфери використання чи домену [40].

У випадку R це статистичне обчислення та аналіз. У розширенні R зазвичай використовується для всіх видів наукових завдань.

R оснащений великим набором функцій, які дозволяють візуалізувати дані, тож користувачі можуть аналізувати дані, моделювати їх за потреби, а потім

створювати графіки. На додаток до вбудованих графічних функцій мови, існують численні доповнення або модулі, які полегшують це.

2.2.1 Сфери та галузі, де використовується R

Оскільки R є потужним і здатним виконувати різноманітні завдання аналізу даних, візуалізації та моделювання, він використовується в різноманітних галузях і секторах[41]. Ось лише деякі з них:

Академічне середовище

Подібно до того, як англійська є лінгва франка у світі, R є домінуючою мовою програмування в багатьох академічних закладах. Його використання також не обмежується статистикою; багато видів досліджень потребують кількісних даних, включаючи кореляційні, експериментальні та описові, і вони відбуваються в різних галузях.

Цифровізація (процес охоплення даних і пов'язаних інструментів) і зростання великих даних торкнулися всіх сфер навчання та досліджень, що призвело до збільшення використання R в академічних умовах [39].

Наприклад, статистичний пакет IBM для соціальних наук колись був провідним програмним забезпеченням для соціальних наук. Зараз R є найкращим вибором з багатьох причин:

– на 100% безкоштовний як для навчальних закладів, так і для студентів;
– сумісний з усіма операційними системами та даними з різних типів файлів;
– забезпечує прозоре та відтворюване дослідження;
– полегшує створення візуалізацій даних.

DataCamp 2013 [42] року щодо R в освіті показав, що 71,1% респондентів вивчали економіку чи бізнес, тоді як лише 10,5% займалися інформатикою, що свідчить про те, що грамотність у роботі з даними та навички мають велике значення.

Data Science.

Поряд з Python, R є важливою мовою у світі науки про дані. За допомогою R професіонали можуть моделювати та аналізувати як структуровані, так і неструктуровані дані, вони також можуть використовувати R для створення інструментів машинного навчання та статистичного аналізу, які допомагають у їхній роботі.

R полегшує роботу з даними з різних джерел, від імпорту до аналізу. Крім того, сама система R і бібліотека CRAN пропонують безліч функцій і інструментів для візуалізації даних, що дозволяє професіоналам легко представляти свої дослідження та висновки в ефектному та легкому для читання форматі [40].

Статистика.

Це само собою зрозуміло, оскільки це мова статистичного програмування, але R є основним інструментом для статистики та статистичних обчислень, зрештою, для цієї мети її розроблено статистиками.

Широкий спектр пакетів підтримує роботу в цій галузі, а саму мову R можна використовувати для розробки програмних засобів, які включають статистичні функції.

Його використання може навіть піти далі. В інтерв'ю [43] інформатик RStudio Джо Ченг зазначає, що R можна використовувати як мову загального призначення для впровадження нових статистичних мов.

Фінанси.

Завдяки своїй гнучкості та здатності виконувати будь-які завдання аналізу даних, не дивно, що R знайшов широке застосування у фінансах. Такі компанії, як ANZ і Bank of America, використовують цю мову для аналізу та моделювання кредитного ризику, фінансової звітності, обробки інвестиційних портфелів та багатьох інших завдань[42].

Спеціальні інструменти, такі як jrvFinance та пакетний пакет Rmetrics, дозволяють тим, хто працює у сфері фінансів, виконувати фінансові обчислення, навіть якщо вони мають обмежений досвід програмування.

Соц.медіа.

З перших днів існування Open Dairy та Bolt соціальні мережі розширили охоплення від кількох технічно підкованих користувачів до практично всіх, хто має смартфон. Сьогодні важко знайти людину, яка не користується соціальними мережами.

Соціальні мережі також є великим бізнесом, бізнесом, який переважно торгує даними. Такі компанії, як Meta (Facebook і Instagram) і TikTok, покладаються на звички користувачів, щоб пропонувати цільову рекламу іншим компаніям[40].

Кожна річ, яку роблять або з якою взаємодіють в соціальних мережах, генерує дані, які можна використовувати для цієї мети, і такі інструменти, як R, є ідеальним способом для компаній соціальних мереж отримати інформацію з масових обсягів даних, які вони збирають, і керувати алгоритмами, які спонукати користувачів повертатися за вмістом, який відповідає їхнім інтересам [39].

2.3 Середовище розробки RStudio

RStudio — це інструмент, який полегшує роботу з R. Це редактор, менеджер версій і інструмент, що підтримує налагодження, створення пакетів, програм і звітів [43].

Умовно розділити робоче вікно на три області:

– **ліва область**: містить вкладки Консоль, Термінал і Фонові завдання;

– **область у верхньому правому куті**: містить вкладки «Середовище», «Історія», «Підключення » та «Посібник»;

– **нижня права область**: включає вкладки «Файли», «Діаграми» , «Пакети» , «Довідка» , «Переглядач» і «Презентація».

2.3.1 Консоль

На цій вкладці спочатку йде інформація про використовувану версію R, а також деякі основні команди. В консолі можна робити, наприклад:

– встановлення та завантаження пакетів R;

– виконання простих або складних математичних операцій;

– присвоєння результату операції змінній;

– імпорт даних;

– створення загальних типів об’єктів R, таких як вектори, матриці або dataframe;

– вивчення даних;

– статистичний аналіз;

– побудова візуалізацій даних.

Однак, коли запускається код безпосередньо в консолі, він не зберігається для подальшого відтворення. Якщо потрібно написати відтворюваний код для вирішення конкретного завдання, потрібно записувати та регулярно зберігати його у файлі сценарію, а не в консолі[43].

Для тестування коду та встановлення пакетів R здебільшого слід використовувати консоль, оскільки їх потрібно інсталювати лише один раз.

2.3.2 Навколишнє середовище

Щоразу, коли визначається нова або повторно призначається існуюча змінна в RStudio, вона зберігається як об’єкт у робочій області та відображається разом із її значенням на вкладці «Середовище» у верхній правій частині вікна RStudio.

Це також стосується більш складних об’єктів, таких як dataframe. Коли імпортуємо дані як dataframe (або створюємо dataframe з нуля), бачимо в робочій області не лише ім’я нового об’єкта, але й значення та тип даних кожного стовпця. Крім того, відобразити ще більше деталей про кожен об’єкт, наприклад його довжину та розмір пам’яті[44].

У наведеному нижче прикладі Рисунок 2.3 створено дві змінні в консолі: `greeting <- "Hello, World!" і my_vector <- c(1, 2, 3, 4)`. Зверніть увагу, як вони відображаються на вкладці «Середовище»:

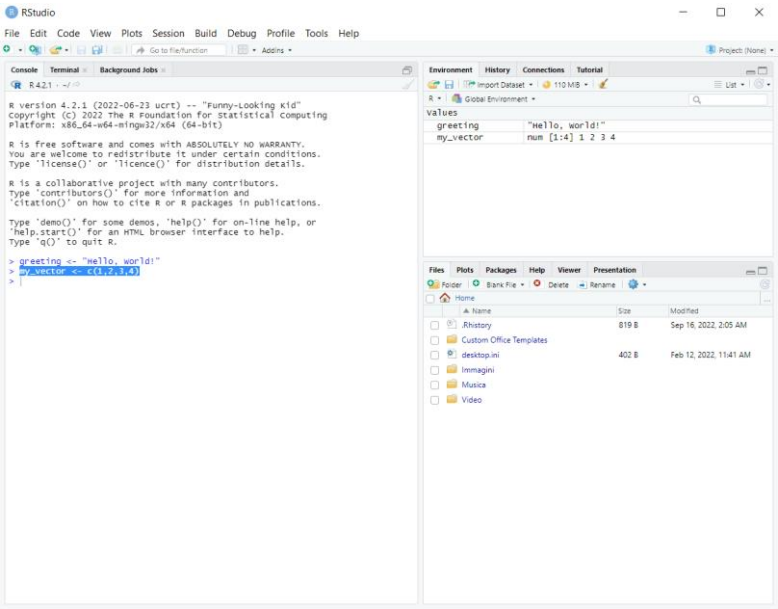

Рисунок 2.5 – Вікно RStudio

Змінити спосіб відображення змінних зі списку на сітку у верхньому правому куті вкладки таким чином:

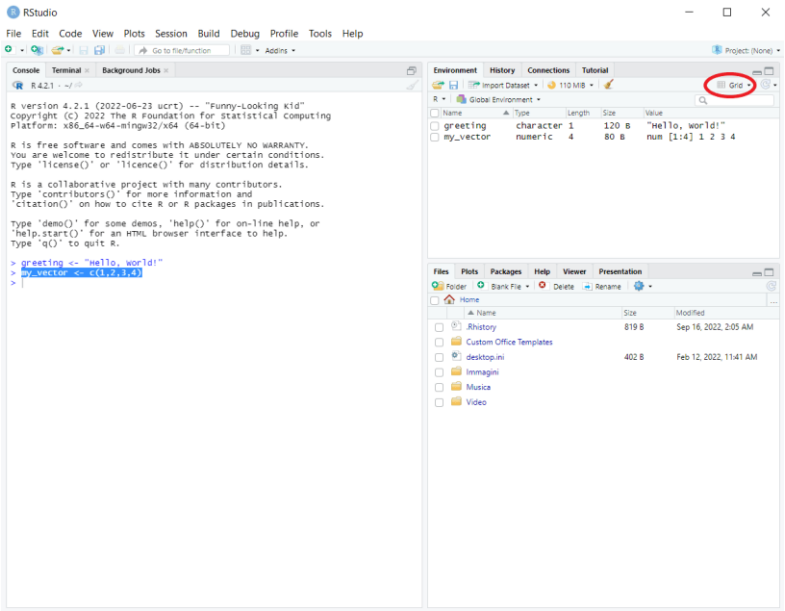

Рисунок 2.6 – Спосіб відображення змінних RStudio

Також, біля кожної змінної також можна бачити довжину та розмір.

У режимі відображення сітки поле з’являється ліворуч від кожної змінної. поставити галочку в будь-якому з цих полів і натиснути піктограму «мітли», щоб видалити відповідні об’єкти з робочої області:

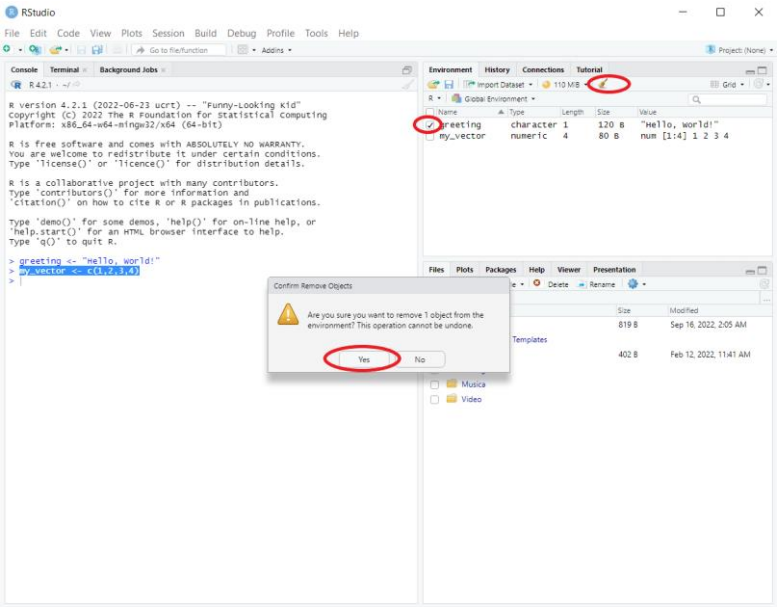

Рисунок 2.7 – Видалення змінної в RStudio

Якщо позначити прапорець ліворуч від стовпця «Назва» та клацнути піктограму «Мітла» або просто клацнути цю піктограму в попередньому режимі відображення («Список»), очиститься робоче середовище, видаливши з нього всі змінні [44].

Інші важливі вкладки

– «Термінал» – для виконання команд із терміналу;

– «Історія» – для відстеження історії всіх операцій, виконаних під час поточного сеансу RStudio;

– «Файли» – для перегляду структури робочої папки, скидання робочої папки, переходу між папка тощо;

– «Графіки» – для попереднього перегляду та експорту створених візуалізацій даних;

– «Пакунки» – щоб перевірити, які пакунки було завантажено, і завантажити або вивантажити пакунки (увімкнувши/вимкнувши поле ліворуч від назви пакунка).

2.4 Технічний аналіз

Технічний аналіз — це інструмент або метод , який використовується для прогнозування ймовірного майбутнього руху ціни цінного паперу. Наприклад, акції або валютної пари – на основі ринкових даних.

Теорія, що лежить в основі обґрунтованості технічного аналізу, полягає в тому, що колективні дії – купівля та продаж – усіх учасників ринку точно відображають всю відповідну інформацію, що стосується цінних паперів, що торгуються, і, отже, постійно призначають цінним паперам справедливу ринкову вартість [45].

Технічні трейдери вважають, що поточна або минула цінова дія на ринку є найнадійнішим індикатором майбутніх цінових дій.

Технічний аналіз використовується не тільки технічними трейдерами. Багато фундаментальних трейдерів використовують фундаментальний аналіз, щоб визначити, чи варто купувати на ринку, але, прийнявши це рішення, потім використовують технічний аналіз, щоб точно визначити хороші рівні початкової ціни покупки з низьким ризиком [46].

2.4.1 Технічні індикатори – індикатори моментуму

Ковзні середні та більшість інших технічних індикаторів в основному зосереджені на визначенні ймовірного напрямку ринку, вгору чи вниз.

Однак існує ще один клас технічних індикаторів, основною метою яких є не стільки визначення напрямку ринку , скільки визначення сили ринку . Ці індикатори включають такі популярні інструменти, як стохастичний осцилятор, індекс відносної сили (RSI), індикатор конвергенції-розбіжності ковзного середнього (MACD).

Вимірюючи силу руху ціни, індикатори моментуму допомагають інвесторам визначити, чи є поточний рух ціни швидше відносно незначною, обмеженою в діапазоні торгівлею, чи фактичною, значною тенденцією. Оскільки індикатори моментуму вимірюють силу тренду, вони можуть служити сигналами раннього попередження про те, що тренд добігає кінця.

Наприклад, якщо цінні папери торгуються в сильному, стійкому висхідному тренді протягом кількох місяців, але потім один або більше індикаторів імпульсу сигналізують про те, що тренд постійно втрачає силу, можливо, настав час подумати про отримання прибутку.

2.4.2 RSI

Індекс відносної сили , розроблений Дж. Уеллсом Уайлдером, є осцилятором імпульсу, який вимірює швидкість і зміну цінових рухів. RSI коливається від нуля до 100. Традиційно RSI вважається перекупленим, коли вище 70, і перепроданим, коли нижче 30. Сигнали можна генерувати, шукаючи розбіжності та коливання невдач. RSI також можна використовувати для визначення загальної тенденції[46].

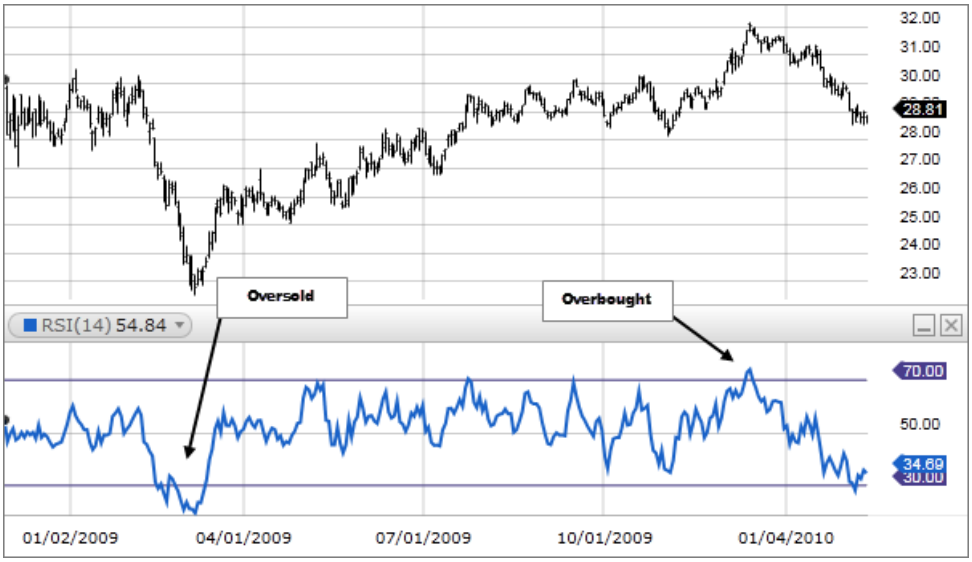

Рисунок 2.8 – Перекупленість і перепроданість [46]

RSI вважається перекупленим, коли вище 70, і перепроданим, коли нижче 30. Ці традиційні рівні також можна скоригувати, якщо необхідно, щоб краще відповідати цінному паперу. Наприклад, якщо цінний папір постійно досягає рівня перекупленості 70, можна змінити цей рівень до 80.

Під час сильних трендів RSI може залишатися в стані перекупленості або перепроданості протягом тривалого часу.

RSI також часто формує моделі діаграм, які можуть не відображатися на базовому ціновому графіку, наприклад подвійні вершини та низи та лінії тренду. Також шукайте підтримку або опір на RSI.

Під час висхідного тренду або бичачого ринку RSI має тенденцію залишатися в діапазоні від 40 до 90, а зона 40-50 виступає як підтримка. Під час спадного тренду або ведмежого ринку RSI має тенденцію залишатися в діапазоні від 10 до 60, а зона 50-60 діє як опір [45]. Ці діапазони змінюватимуться залежно від налаштувань RSI та потужності основного тренду цінного паперу чи ринку.

Якщо базові ціни досягають нового максимуму або мінімуму, що не підтверджується RSI, це розбіжність може сигналізувати про розворот ціни. Якщо RSI досягає нижчого максимуму, а потім слідує рух вниз нижче попереднього мінімуму, сталася помилка верхнього коливання. Якщо RSI досягає вищого мінімуму, а потім слідує рухом угору вище попереднього максимуму, сталася помилка нижнього коливання.RSI є досить простою формулою, але її важко пояснити без прикладів. Основна формула:

$$RSI = 100 - \left[\frac{100}{1 - \frac{\text{середній приріст}}{\text{середня втрата}}}\right] \qquad (2.5)$$

Середній прибуток або збиток, який використовується в цьому розрахунку, є середнім відсотковим приростом або збитком протягом оглядового періоду. У формулі використовується додатне значення середньої втрати.

Чому RSI важливий:

– трейдери можуть використовувати RSI для прогнозування цінової поведінки цінного паперу;

– це може допомогти трейдерам перевірити тренди та розвороти трендів;

– це може вказувати на перекуплені та перепродані цінні папери;

– він може надавати короткостроковим трейдерам сигнали купівлі та продажу;

– це технічний індикатор, який можна використовувати разом з іншими для підтримки торгових стратегій.

Розбіжність RSI виникає, коли ціна рухається в протилежному напрямку від RSI. Іншими словами, діаграма може відображати зміну моментуму перед відповідною зміною ціни[45].

Бичача дивергенція виникає, коли RSI показує перепроданість, за якою слідує вищий мінімум, який з'являється разом із нижчими мінімумами ціни. Це може вказувати на зростання бичачого імпульсу, і прорив вище території перепроданості може бути використаний для запуску нової довгої позиції .

Ведмежа дивергенція виникає, коли RSI створює показники перекупленості, за якими слідує нижчий максимум, який з'являється разом із вищими максимумами ціни.

Як можна бачити на рисунку 2.9, було виявлено бичачу дивергенцію, коли RSI сформував вищі мінімуми, оскільки ціна сформувала нижчі мінімуми. Це був дійсний сигнал, але розбіжності можуть бути рідкісними, коли акція має стабільний довгостроковий тренд. Використання гнучких показників перепроданості або перекупленості допоможе визначити більше потенційних сигналів.

Рисунок 2.9 – Бичача дивергенція. Ціна падає RSI зростає [46]

2.5 Технології – Santiment

Santiment — це криптокомпанія, яка надає канали даних і аналіз ринку для криптоіндустрії. Вони спрямовані на те, щоб допомогти криптоінвесторам приймати обґрунтовані рішення та уникати дорогих помилок[34].

Santiment збирає, аналізує та розповсюджує криптодані в режимі реального часу, надаючи своїм користувачам практичну інформацію, а саме щоб поєднати мережеву аналітику з даними в реальному часі, отриманими з соціальних мереж, і загальними настроями криптовалютного ринку. Окрім каналів даних, Santiment також пропонує набір комплексних інструментів.

Платформа дозволяє відстежувати, візуалізувати та аналізувати криптовалютні ринки.

Компанії можуть приймати більш обґрунтовані рішення, розуміючи, як люди використовують і взаємодіють з технологією блокчейн. Переваги цього процесу очевидні, коли справа доходить до інвестування грошей на ринку[34].

Аналіз даних також стає все більш важливим для цілей відповідності. Оскільки все більше країн регулюють індустрію блокчейну, підприємства повинні продемонструвати, що вони дотримуються всіх чинних законів.

2.5.1 Sanbase

Sanbase — це веб-платформа Santiment, яка надає користувачам доступ до всіх даних Santiment. Ми можемо згадати ціни в реальному часі, історичні дані та активність у соціальних мережах як приклади даних, які можна збирати[35].

З Sanbase дає можливість:

- отримувати огляд криптовалютного ринку за допомогою рейтингу ринкової капіталізації Santiment і діаграм цін;
- відстежувати ефективність активів за допомогою історії цін і графіків змін цін;
- бути в курсі останніх новин і активності в соціальних мережах за допомогою віджетів «Стрічка новин» і «Соціальні настрої»;
- отримати статистичні дані про конкретні активи за допомогою користувальницьких індикаторів і сигнальних сповіщень Santiment.

Sanbase є ідеальною платформою як для початківців, так і для досвідчених криптоінвесторів, які хочуть залишатися попереду. Він також функціонує як база даних настроїв натовпу для криптовалютних проектів, де можна знайти

інформацію про монету, таку як команда, що стоїть за нею, її історія, аналіз цін тощо.

2.6 Метрики

Метрики є числовими показниками, які використовуються для оцінки ефективності моделей машинного навчання. Вони дозволяють кількісно виміряти якість прогнозів, зроблених моделлю, та порівнювати різні моделі між собою. Метрики використовуються для об'єктивного оцінювання того, наскільки добре модель виконує свою задачу.

2.6.1 Точність

Найпростішою метрикою у задачі бінарної класифікації є точність (accuracy), яка обчислюється за формулою:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{TP + TN + FP + \text{FN}} \tag{2.6}$$

де:

TP – кількість правильно класифікованих позитивних екземплярів;

TN – кількість правильно класифікованих негативних екземплярів;

FP – кількість неправильно класифікованих позитивних екземплярів;

FN – кількість неправильно класифікованих негативних екземплярів.

Для роботи з незбалансованими наборами даних ця метрика не підходить через свою нечутливість до розподілу класів. При роботі з незбалансованими даними, коли один клас переважає над іншим, модель може досягти високої точності, просто передбачаючи більш представлений клас у кожному випадку.

Наприклад, якщо маємо 95% негативних екземплярів і 5% позитивних екземплярів, то модель може завжди передбачати негативний клас і досягти точності 95%, незалежно від її здатності до правильного виявлення позитивних екземплярів.

2.6.2 Влучність і повнота

Метрики влучність (precision) і повнота (recall) є двома важливими метриками для оцінки ефективності бінарної класифікації. Влучність вимірює точність моделі в прогнозуванні позитивного класу. Вона показує, яка частина позитивно класифікованих зразків є дійсно позитивними. Влучність вказує, наскільки мало модель допускає помилкових позитивних класифікацій. В контексті шахрайських транзакцій, влучність показує, яка частина зазначених моделлю шахрайських транзакцій є дійсно шахрайськими.

Повнота, також відома як чутливість (sensitivity) або TPR, вимірює, яку частку дійсно позитивних зразків здатна виявити модель. Вона показує, наскільки ефективно модель виявляє позитивні зразки. В контексті шахрайських транзакцій, повнота вказує, яка частина дійсних шахрайських транзакцій була коректно виявлена моделлю.

У бінарній класифікації шахрайських транзакцій, важно досягти високої влучності та високої повноти. Висока влучність гарантує, що помилкові позитивні класифікації шахрайських транзакцій будуть мінімізовані, забезпечуючи, що більшість класифікованих шахрайських транзакцій є дійсно шахрайськими. З іншого боку, висока повнота важлива для забезпечення того, що жодна дійсна шахрайська транзакція не буде пропущена і буде виявлена моделлю.

2.6.3 F1 міра

F1 міра є середнім гармонійним між влучністю і повнотою. Вона використовується як метрика для оцінки ефективності бінарної 43 класифікації, особливо в ситуаціях, коли потрібно збалансувати важливість точності та повноти. F1 міра об'єднує якість класифікації в обох напрямках, тобто якість визначення як позитивних, так і негативних зразків. Вона обчислюється за формулою (2.7):

$$\mathrm{F1} = 2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \qquad (2.7)$$

F1 міра приймає значення від 0 до 1, де 1 вказує на найкращу можливу класифікацію. Така метрика особливо корисна в ситуаціях, коли важливі як точність, так і повнота, і неможливо вибрати одну з них в якості основної метрики.

Наприклад, у випадку виявлення шахрайських транзакцій, F1-міра допомагає забезпечити баланс між точністю (мінімізацією помилкових позитивних класифікацій) та повнотою (максимізацією виявлення шахрайських транзакцій).

Враховуючи F1-міру, можна оцінити загальну ефективність класифікатора, забезпечити баланс між точністю та повнотою, а також порівняти різні моделі машинного навчання або налаштування моделі для покращення результатів.

2.6.4 AUC-ROC

AUC-ROC Метрика AUC-ROC використовується для оцінки якості класифікаційної моделі, зокрема для бінарних класифікаційних задач. ROC (Receiver Operating Characteristic) – це графічна характеристика, яка показує залежність між чутливістю – TPR і специфічністю – FPR при різних порогових значеннях класифікації. Нижче наведені формули (2.8) і (2.9) для визначення TPR та FPR.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.8}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{2.9}$$

AUC-ROC обчислює площу під цією кривою ROC. Значення AUC-ROC може бути в діапазоні від 0 до 1, де 0 означає неправильну класифікацію для всіх екземплярів, а 1 означає ідеальну класифікацію.

Для розрахунку AUC-ROC спочатку обчислюються значення TPR та FPR для різних порогових значень. TPR відображає відношення правильно класифікованих позитивних екземплярів до загальної кількості позитивних екземплярів. FPR відображає відношення неправильно класифікованих негативних екземплярів до загальної кількості негативних екземплярів. Потім ці значення TPR і FPR використовуються для побудови кривої ROC. Далі обчислюється площа під цією кривою, що і є значенням AUC-ROC.

Висновки до розділу 2

У другому розділі представлено ключові технології та методи для розв'язання поставленої задачі. Розглянуто регресійні методи, які є фундаментальними для статистичного аналізу та прогнозування. Мова програмування R та середовище розробки RStudio виокремлюються як основні інструменти для статистичних обчислень і аналізу даних. Технічний аналіз вводиться як важливий елемент фінансового аналізу, зосереджений на вивченні історичних даних ринку.

Також розглянуто існуючі системи аналізу новин, Santiment представлено як інноваційний інструмент для аналізу соціальних медіа та фінансових ринків, забезпечуючи глибший вгляд у ринкові тенденції та настрої.

Розділ демонструє, як ці технології можуть бути інтегровані для ефективного рішення комплексних задач. Загалом, цей розділ надає огляд різних технологій та методів, які можуть бути використані для аналізу та класифікації

3 ПРОГРАМНА РЕАЛІЗАЦІЯ РЕГРЕСИВНИХ МОДЕЛЕЙ

3.1 Підготовка даних

3.1.1 Опис джерела даних

Для роботи було обрано використовувати історичні дані новин з сервісу Crypto News API [36]. Цей сервіс спеціалізується на наданні чистих даних крипто сегменту, а також суворий процес розділення, який гарантує, що всі новини стосуються саме вказаної криптовалюти, а також вони надають до кожної новини власну систему оцінки настрою.

Сентимент базуються на аналізі ключових слів. Для кожної новини є 3 типи позначок, а саме: позитивна, негативна або нейтральна. При визначенні настрою враховується як назву, так і опис. Крім того, розглядається важливість і частоту конкретних ключових слів.

Одне з головних переваг є те що система також враховує такі ключові слова зі сленгу Уолл-стріт, як «бичачий», «ведмежий», «голубиний», «яструбиний відскок мертвої кішки», «коротке стискання» та багато інших...

Данні віддаються по денно в форматі JSON.


```
{
  "data": [
    {
      "news_url": "https://cryptoticker.io/en/solana-ecosystem-top-10-solana-coins/",
      "image_url": "https://crypto.snapi.dev/images/v1/l/6/crypto7-414040.jpg",
      "title": "Solana Ecosystem: Top 10 Solana Coins That Could Make You a Millionaire in 2024",
      "text": "We had already predicted in our last SOL price prediction that SOL could reach $100 soon. This article is all about the top 10 Solana coins",
      "source_name": "CryptoTicker",
      "date": "Fri, 22 Dec 2023 11:05:39 -0500",
      "topics": [

      ],
      "sentiment": "Positive",
      "type": "Article",
      "tickers": [
        "SOL"
      ]
    },
```


Рисунок 3.1– Вигляд запиту усіх новин

API дає можливість вибирати дані по тікету (BTC, USD та інші.), рангу, даті, сентименту, типу, темі та інші. Також, використовуючи преміум підписку, можна

отримувати історичні дані, фільтрувати їх та конвертувати в інші формати, окрім JSON.

Також дані збираються більше ніж з 30 джерел. В які входять як великі так і маленькі видавці.

Для роботи буде використано запит для отримання даних по кількості позитивних, негативних і нейтральних новин за цей день, а саме:

«api/v1/stat?&tickers={1}&date={2}&page={3}&token={4}»

– “tickers” – назва активу;

– “date” – інтервал дат;

– “page” – номер сторінки (запит повертає не більше 30 днів за один запит);

– “token” – ключ користувача.

Результат запиту можна побачити на рисунку 3.2


```
{
  "total": {
    "BTC": {
      "Total Positive": 2937,
      "Total Negative": 573,
      "Total Neutral": 553,
      "Sentiment Score": 0.873
    }
  },
  "data": {
    "2023-12-22": {
      "BTC": {
        "Neutral": 14,
        "Positive": 62,
        "Negative": 10,
        "sentiment_score": 0.907
      }
    },
    "2023-12-21": {
      "BTC": {
        "Neutral": 17,
        "Positive": 109,
        "Negative": 16,
        "sentiment_score": 0.982
      }
```


Рисунок 3.2 – Запит настроїв по дням

Настрої сервіс визначає за ключовими словами і загальним ефектом новини на певний актив та сферу впливу.

3.1.2 Обробка даних

У роботі використано історичні данні які включають в себе поділені на 3 типу новини, а саме:

– позитивні;

– негативні;

– нейтральні.

Далі розглянемо код створеного застосунку. Він складається з ряду кроків, що включають установку та завантаження необхідних пакетів, отримання історичних даних цін на Bitcoin з використанням Yahoo Finance, виконання HTTP-запитів для отримання даних про настроїв на ринку з веб-сервісу cryptonews-api.com, та обробку цих даних для подальшого аналізу.

У початковому етапі програма інсталює та завантажує необхідні пакети для обробки фінансових даних, роботи з HTTP-запитами та обробки JSON-даних.

```
install.packages(c("httr",
"jsonlite","quantmod","caret","pROC","ROCR","class","e1071")
library(e1071)
library(class)
library(httr)
library(jsonlite)
library(quantmod)
library(caret)
library(pROC)
library(ROCR)
library(dplyr)
```

Далі вона отримує історичні дані цін на Bitcoin за певний період, що дозволяє здійснювати аналіз динаміки цін.

```
# Завантаження даних з біткоїну з Yahoo Finance
getSymbols("BTC-USD", src = "yahoo", from = "2020-12-25", to = Sys.Date())
# Створення змінної зростання/падіння ціни
price_changes <-diff(Cl(`BTC-USD`)) > 0
```

	BTC-USD.Open	BTC-USD.High	BTC-USD.Low	BTC-USD.Close	BTC-USD.Volume	BTC-USD.Adjusted
2024-01-21	41671.49	41855.37	41497.01	41545.79	9344043642	41545.79
2024-01-22	41553.65	41651.21	39450.12	39507.37	31338708143	39507.37
2024-01-23	39518.71	40127.35	38521.89	39845.55	29244553045	39845.55
2024-01-24	39877.59	40483.79	39508.80	40077.07	22359526178	40077.07
2024-01-25	40075.55	40254.48	39545.66	39933.81	18491782013	39933.81
2024-01-26	39936.82	41450.40	39825.94	41017.56	22273878016	41017.56

Рисунок 3.3 – Формат даних BTC-USD

Ключовим елементом є функція request, яка виконує HTTP-запит на веб-сервіс cryptonews-api.com для отримання даних про настроїв ринку стосовно Bitcoin. Результати запиту обробляються та аналізуються, а дані про настрій перетворюються у відносні відсотки нейтрального, позитивного та негативного настрою.

Сторінка для запиту сформована по шаблону який був описаний в 2.2 цієї роботи

```
request <- function(page){
url<-paste0("Сторінка запиту")
  response <- GET(url)
  # Перевірка успішності запиту
  if (http_type(response) == "application/json") {
    # Парсинг JSON
    json_content <- content(response, "text", encoding = "UTF-8")
    parsed_data <- fromJSON(json_content)
    # Перегляд даних
    return(parsed_data)
  } else {
    warning("Помилка запиту")
  }
}
```

У заключній частині коду дані про настрій додаються до загального фрейму даних, створеного на основі історичних цін на Bitcoin. Таким чином, результатом роботи програми є звіт, що містить історичні дані цін на Bitcoin та дані про настроїв ринку стосовно цієї криптовалюти, що може бути використано для подальшого аналізу та розробки стратегій торгівлі або інвестування.

```
parsed_data <-request(1)
news <- data.frame(Date = as.Date(character()), Neutral = double(), Positive
= double(), Negative  = double())
i<-1
while(i <= parsed_data$total_pages){
  for (dateN in names(parsed_data$data)) {
    date <- parsed_data$data[[dateN]]$BTC
    sum <-date$Neutral+date$Positive+date$Negative
    temp <- data.frame(
      Date = as.Date(dateN),
      Neutral = date$Neutral/sum,
      Positive = date$Positive/sum,
      Negative = date$Negative/sum
    )
    news <- rbind(news, temp)
  }
  parsed_data<-request(i)
  i<-i+1
}

price_changes_df  <-  data.frame(Date  =  index(price_changes),  dPrice  =
price_changes$`BTC-USD.Close`)
```

Таким чином, результатом виконання цього коду буде дата фрейм даних, який містить історичні дані руху цін на Bitcoin, а також дані про настрій учасників ринку стосовно цієї криптовалюти за кожну дату. Ця інформація може бути корисною для подальшого аналізу та прийняття рішень щодо інвестування чи торгівлі Bitcoin.

	Date	Neutral	Positive	Negative	dPrice
1145	2024-01-21	0.20338983	0.4923729	0.3042373	0
1146	2024-01-22	0.08187135	0.3359649	0.5821637	0
1147	2024-01-23	0.07798165	0.4661468	0.4558716	1
1148	2024-01-24	0.13966480	0.6710056	0.1893296	1
1149	2024-01-25	0.08720930	0.5488372	0.3639535	0
1150	2024-01-26	0.10280374	0.7255140	0.1716822	1

Рисунок 3.4 – Перший набір підготовлених даних

У контексті розширення функціоналу було додано обчислення Relative Strength Index (RSI), яке використовується для аналізу динаміки цін на Bitcoin. Це призвело до створення двох наборів даних для подальшого порівняльного аналізу. Перший набір містить настрої новин Bitcoin разом з відповідними значеннями RSI для кожної дати, тоді як другий набір складається лише з настроїв без включення RSI.

	Date	Neutral	Positive	Negative	dPrice	rsi
1145	2024-01-21	0.20338983	0.4923729	0.3042373	0	0.4305913
1146	2024-01-22	0.08187135	0.3359649	0.5821637	0	0.3568788
1147	2024-01-23	0.07798165	0.4661468	0.4558716	1	0.3759654
1148	2024-01-24	0.13966480	0.6710056	0.1893296	1	0.3893274
1149	2024-01-25	0.08720930	0.5488372	0.3639535	0	0.3838502
1150	2024-01-26	0.10280374	0.7255140	0.1716822	1	0.4773252

Рисунок 3.5 – Другий набір підготовлених даних

```
s<-RSI(`BTC-USD`$`BTC-USD.Close`)
s$rsi<-s$rsi/100
rsi <- data.frame(Date = index(s), rsi = s$rsi)
rsi <- rsi %>%
  slice((14 + 1):n())
rsiData <- merge(news, rsi, by = "Date", all.x = TRUE)
```

У контексті класифікації руху ціни, включення RSI може бути корисним, оскільки він надає додаткові дані про стан ринку. Разом з історичними цінами, значення RSI можуть допомогти моделі в ідентифікації періодів перекупленості та перепроданості, що в свою чергу може вплинути на подальшу динаміку цін. Включення RSI дозволяє моделі здійснювати більш точні прогнози та краще

управляти ризиком шляхом розрізнення потенційних змін у напрямку цін на Bitcoin.

Далі розглянуто злиття коду в один датафрейм для подальшох роботи

```
merged_data <- merge(news, price_changes_df, by = "Date", all.x = TRUE)
```

Перший крок коду включає злиття двох наборів даних: **news** та **price_changes_df**. Це злиття виконується за стовпцем "Date", який є спільним для обох наборів даних, з використанням параметра **all.x = TRUE**, що забезпечує збереження всіх записів з першого набору даних (**news**) у випадку, якщо для деяких з них не знайдено відповідностей у другому наборі даних. Це дозволяє врахувати всі новинні дані, незалежно від наявності відповідних записів про зміни цін.

```
merged_data <- merge(merged_data, rsi, by = "Date", all.x = TRUE)
```

Другий крок розширює отриманий у першому кроці набір даних, включаючи до нього інформацію з третього набору даних **rsi**, який також зливається за стовпцем "Date". Параметр **all.x = TRUE** використовується знову, щоб гарантувати, що всі рядки з попереднього результату злиття залишаться в кінцевому наборі даних, зберігаючи цілісність даних.

```
colnames(merged_data)[colnames(merged_data) == "BTC.USD.Close"] <- "dPrice"
```

Після злиття наборів даних код вносить зміни у назви стовпців, конкретно змінюючи назву стовпця "BTC.USD.Close" на "dPrice". Ця операція спрощує ідентифікацію стовпця для подальшого аналізу, роблячи назву стовпця більш інтуїтивно зрозумілою.

```
merged_data$dPrice <- as.numeric(merged_data$dPrice)
```

Останнім кроком є перетворення даних у стовпці "dPrice" із їхнього поточного формату в числовий формат за допомогою функції **as.numeric**. Ця операція необхідна для того, щоб можна було проводити кількісний аналіз цінових

змін, включаючи обчислення статистичних показників, графічне представлення даних та інші аналітичні процедури.

3.1.3 Поділ вибірки

У роботі представлено методологію поділу агрегованого набору даних на тренувальний та тестовий сегменти. Ініціація процесу розпочалася з встановлення початкового значення для генератора випадкових чисел за допомогою функції **set.seed(123)**, що забезпечує відтворюваність експериментальних даних.

```
sample_index <- sample(1:nrow(merged_data), 0.7 * nrow(merged_data))
```

Далі була виконана випадкова вибірка індексів за допомогою функції **sample**, де аргументи функції задавали діапазон від 1 до загальної кількості рядків в агрегованому наборі даних (**nrow(merged_data)**) та розмір вибірки, еквівалентний 70% від загальної кількості рядків. Вказаний підхід дозволяє сформувати тренувальний набір даних, який балансує між достатньою кількістю даних для ефективного навчання моделі та збереженням адекватної кількості даних для її подальшої верифікації.

```
train_data <- merged_data[sample_index, ]
test_data <- merged_data[-sample_index, ]
```

На наступному етапі було сформовано тренувальний набір даних **train_data** шляхом вибору рядків з агрегованого набору даних (**merged_data**) відповідно до отриманих індексів. Симетрично, тестовий набір даних **test_data** було сформовано з рядків, які не були включені в тренувальний набір, що досягається застосуванням від'ємної індексації.

```
test_data$dPrice <- as.factor(test_data$dPrice)
```

Завершальним кроком аналізу було перетворення значень стовпця **dPrice** у тестовому наборі даних на факторний тип (**as.factor**), що є критично важливим для аналітичних досліджень, а саме для подальшого використання функції

confusionMatrix, а також у контексті класифікаційних задач, де категоріальні змінні відіграють ключову роль у моделюванні та інтерпретації даних.

3.1.4 Створення хелперів

Для того щоб в подальшому використовувати повторно формулу предикторів і не копіювати в кожну з моделей винесемо в окрему зміну

```
predictors <- c("Positive", "Negative", "Neutral","rsi")
```

Спосіб полягає у створенні вектора **predictors**, який містить назви змінних-предикторів, що будуть використані в моделі. У даному випадку, предикторами є "Positive", "Negative", та "Neutral", "rsi". Ці назви представляють собою змінні, які впливають на змінну відгуку **dPrice** у статистичній моделі.

```
formula <- as.formula(paste("dPrice ~", paste(predictors, collapse = " +
")))
```

Сама формула включає використання вектора **predictors** для динамічного створення формули, яка використана в функціях моделювання. Формула створюється за допомогою функції **paste()**, яка об'єднує елементи вектора **predictors** у один рядок з операторами "+", відповідно до синтаксису формули R, де **~** розділяє змінну відгуку та предиктори. Результат перетворюється на формулу за допомогою **as.formula()**.

Для оцінки моделей створено функцію **evaluate_model()**, вона є інструментом для оцінювання продуктивності статистичних моделей або моделей машинного навчання. На вхід приймає два параметри: **true_values**, які представляють справжні мітки класу, і **predicted_values**, що відповідають за прогнозовані моделлю мітки. Центральною частиною функції є визначення матриці помилок, яка лягає в основу розрахунку ключових метрик оцінювання моделі, включаючи точність (accuracy), точність (precision), повноту (recall) та F1-оцінку.

```
evaluate_model <- function(true_values, predicted_values) {
  conf_matrix        <-        confusionMatrix(as.factor(predicted_values),
as.factor(true_values))
  accuracy <- conf_matrix$overall[['Accuracy']]
  precision <- conf_matrix$byClass[['Pos Pred Value']]
  recall <- conf_matrix$byClass[['Sensitivity']]
  f1_score <- conf_matrix$byClass[["F1"]]
 # Повернення списку метрик
  return(c(Accuracy=accuracy,   Precision=precision,   Recall=recall,   `F1
Score`=f1_score))
}
```

Матриця помилок, генерована за допомогою **confusionMatrix** з пакету **caret**, дозволяє не тільки оцінити кількість правильних і неправильних передбачень, але й надає детальну статистику про продуктивність моделі.

Точність моделі (accuracy) відображає загальний відсоток правильно класифікованих випадків, тоді як точність (precision) і повнота (recall) надають інформацію про якість класифікації позитивного класу. F1-оцінка, що є гармонічним середнім між точністю і повнотою, використовується для оцінки балансу між ними, що є особливо корисним в умовах нерівномірного розподілу класів.

Після обчислення, функція повертає список метрик, який може бути використаний для детального аналізу та порівняння моделей. Цей підхід дозволяє дослідникам та аналітикам ефективно оцінювати продуктивність різних моделей на основі стандартизованих критеріїв, сприяючи об'єктивному вибору найбільш адекватної моделі для конкретної задачі аналізу даних.

3.2 Створення моделей

3.2.1 Логіт модель

Модель логістичної регресії навчається за допомогою функції **glm()** (Generalized Linear Models), де в якості аргументів передаються формула, що описує залежність між змінною відгуку та предикторами, та набір даних для тренування (**train_data**). Формула, визначена в попередніх кроках, використовується для вказівки залежності змінної відгуку **dPrice** від предикторів "Positive", "Negative" та "Neutral".

```
model <- glm(formula, data = train_data, family = "binomial")
```

Після навчання моделі виконується предбачення на тестовому наборі даних за допомогою функції **predict()**. Предбачені значення (**predictions**) представляють собою неперервні оцінки, що відображають передбачену ймовірність належності до класу 1.

Для перетворення цих неперервних оцінок у бінарні класи використовується порогове значення 0.5, де результати вище порога класифікуються як 1 (позитивний клас), а нижче - як 0 (негативний клас).

```
predictions <- predict(model, newdata = test_data)
predicted_classes <- ifelse(predictions > 0.5, 1, 0)
```

Фінальний етап передбачає оцінку ефективності моделі на основі тестових даних. Для цього використовується раніше визначена функція **evaluate_model**, яка приймає як аргументи реальні мітки класу (**test_data$dPrice**) та предбачені класи (**predicted_classes**).

Результатом виконання цієї функції є набір метрик, таких як точність (Accuracy), точність (Precision), повнота (Recall) та F1-оцінка, що дозволяє оцінити якість моделі.

```
model1_metrics<-evaluate_model(test_data$dPrice,predicted_classes)
```

```
 Accuracy Precision    Recall  F1 Score
0.7855072 0.7874396 0.8445596 0.8150000
```

Рисунок 3.6 – Результат логістичної моделі з RSI

```
 Accuracy Precision    Recall  F1 Score
0.7768116 0.7636364 0.8704663 0.8135593
```

Рисунок 3.7 – Результат логістичної моделі без RSI

Розглядаючи результати метрик двох моделей логістичної регресії, для моделі з RSI точність (Accuracy) складає 0.7855, влучність (Precision) — 0.7874, повнота (Recall) — 0.8446, і F1-оцінка (F1 Score) — 0.8150. Ці результати вказують на те, що модель з RSI досить добре ідентифікує позитивні випадки, що відображається у високому відклику та F1-оцінці, яка є гармонійним середнім між прецизійністю та відкликом.

Модель логістичної регресії без використання RSI показала трохи нижчу точність 0.7768, влучність 0.7636, але повнота 0.8705 і порівняно високу F1-оцінку 0.8136. Висока повнота у моделі без RSI може свідчити про її здатність ефективно ідентифікувати позитивні класи, однак збільшення кількості помилково позитивних результатів може впливати на влучність.

Загалом, додавання RSI як предиктора у модель логістичної регресії злегка покращило точність та F1-оцінку, що робить цей індикатор корисним для збільшення загальної ефективності моделі у цьому конкретному випадку аналізу даних. Однак, це також призвело до незначного зниження повноти.

3.2.2 Модель KNN

Перед початком створення моделі і отримання результатів потрібно підібрати правильний k, для цього спочатку ініціалізується вектор **k_values**, що містить послідовність значень k від 1 до 400, та вектор **accuracy_scores** для зберігання точності моделі при різних k

```
k_values <- 1:400
accuracy_scores <- numeric(length(k_values))
```

Далі цикл for проходить через кожне значення k, використовуючи його для навчання моделі KNN за допомогою функції **knn**. Для навчання моделі використовуються обрані ознаки (predictors) з навчального набору даних **train_data** і тестового набору **test_data**, а також мітки класів **train_data$dPrice**, що, представляють ціновий діапазон цін.

```
for (i in 1:length(k_values)) {
  k <- k_values[i]
  # Навчання моделі KNN з поточним значенням k
  knn_model <- knn(train = train_data[, predictors],
                   test = test_data[, predictors],
                   cl = train_data$dPrice,
                   k = k)
    # Оцінка точності на валідаційній вибірці і збереження її
  accuracy_scores[i] <- mean(knn_model == test_data$dPrice)
}
```

Після навчання моделі для кожного значення k обчислюється точність на тестовій вибірці, що дозволяє оцінити якість моделі. Точність обчислюється як середнє значення логічних порівнянь між прогнозованими моделлю мітками і справжніми мітками даних з **test_data$dPrice**. Результати зберігаються у векторі **accuracy_scores**.

```
best_k <- k_values[which.max(accuracy_scores)]
cat("Найкраще значення k:", best_k, "\n")
```

Після завершення ітерацій знаходиться максимальне значення у векторі точності, що відповідає оптимальному значенню k, за яким модель демонструє найкращу точність. Оптимальне значення k використовується для фінального навчання моделі KNN.

```
> cat("Найкраще значення k:", best_k, "\n")
Найкраще значення k: 11
```

Рисунок 3.8 – Результат пошуку найкращого k з RSI

```
> cat("Найкраще значення k:", best_k, "\n")
Найкраще значення k: 44
```

Рисунок 3.9– Результат пошуку найкращого k без RSI

У завершенні, модель оцінюється за допомогою функції **evaluate_model**, яка, порівнює прогнозовані значення з фактичними і виводить метрики якості моделі. Результати оцінки виводяться за допомогою **print(model2_metrics)**.

```
> print(model2_metrics)
 Accuracy Precision    Recall  F1 Score
0.7536232 0.7903226 0.7616580 0.7757256
```

Рисунок 3.10 – Приклад погано підібраного k = 4

```
> print(model2_metrics)
 Accuracy Precision    Recall  F1 Score
0.7942029 0.8144330 0.8186528 0.8165375
```

Рисунок 3.11 – Результат моделі k = 11 з RSI

```
 Accuracy Precision    Recall  F1 Score
0.7681159 0.7627907 0.8497409 0.8039216
```

Рисунок 3.12 Результат моделі k = 44 без RSI

На рисунках 3.10-3.12 представлені результати моделей k-найближчих сусідів (KNN) з різними наборами предикторів. У першій моделі, де було вибрано значення k = 4 і використовувалися всі доступні предиктори, точність (Accuracy) складає 0.7536, точність позитивного класу (Precision) — 0.7903, повнота (Recall) — 0.7616, та F1-оцінка — 0.7757. Результати вказують на достатньо збалансовану модель з непоганою здатністю до класифікації.

Друга модель KNN з оптимальним k = 11 та використанням індексу відносної сили (RSI) як одного з предикторів показала покращення по більшості метрик: точність зросла до 0.7942, точність позитивного класу до 0.8144, повнота значно підвищилася до 0.8165, а F1-оцінка досягла 0.8039. Такі показники свідчать про високу ефективність моделі у виявленні позитивних класів та її здатність узагальнювати навчальні дані.

Третя модель KNN, з k = 44, але без використання RSI як предиктора, продемонструвала зниження у всіх метриках: точність знизилася до 0.7681, точність позитивного класу до 0.7627, повнота до 0.8497, а F1-оцінка до 0.8039.

Загалом, можна зробити висновок, що KNN із включенням RSI показує кращі результати, це вказує на важливість RSI в наборі предикторів для цієї конкретної задачі прогнозування, що підкреслює значення цього параметра для аналізу фінансових часових рядів та значимість вибору оптимального числа найближчих сусідів.

3.2.3 Модель SVM

Створюється модель SVM використовуючи функцію **svm** з пакета **e1071**. Функція **svm** приймає формулу і набір навчальних даних **train_data**, де формула **formula** визначає залежну змінну (у цьому випадку - **dPrice**) як функцію від незалежних змінних, або предикторів, таких як "Positive", "Negative", "Neutral", і "RSI", в залежності від контексту.

```
svm_model <- svm(formula, data = train_data)
svm_pred <- predict(svm_model, newdata = test_data[, predictors])
```

Після навчання моделі, виконується прогнозування на тестовому наборі даних **test_data** за допомогою функції **predict**, яка застосовується до навченої моделі SVM і нового набору даних, що містить лише вибрані предиктори.

Результатом виклику **predict** є вектор прогнозованих значень **svm_pred**, які відповідають прогнозованим класам або значенням відповідно до моделі SVM на тестових даних.

```
# Підготовка до оцінки
svm_predicted_classes <- ifelse(svm_pred > 0.5, 1, 0)
# Оцінка моделі
model3_metrics = evaluate_model(test_data$dPrice,svm_predicted_classes)
```

Далі здійснюється підготовка до оцінки ефективності моделі машини опорних векторів (SVM) та її наступна оцінка на тестових даних. Напочатку,

отримані від SVM неперервні прогнозовані значення **svm_pred** конвертуються в бінарні класи за допомогою функції **ifelse**, де кожне прогнозоване значення порівнюється з порогом 0.5. Якщо прогнозоване значення перевищує 0.5, воно вважається класом 1, в іншому випадку — класом 0. Це перетворення дає змогу отримати вектор **svm_predicted_classes**, який містить бінарні класи, що відповідають кожному випадку у тестовому наборі даних.

Далі, оцінка моделі проводиться за допомогою функції **evaluate_model**, яка приймає фактичні класи з тестового набору даних **test_data$dPrice** та прогнозовані класи **svm_predicted_classes**.

Функція **evaluate_model** використовує матрицю помилок для обчислення таких метрик як точність (Accuracy), влучність (Precision), повнота (Recall) та F1-оцінка (F1 Score), які відображають загальну ефективність моделі. Значення цих метрик зберігаються у змінній **model3_metrics**.

```
> print(model3_metrics)
 Accuracy Precision    Recall  F1 Score
0.7913043 0.7951220 0.8445596 0.8190955
```

Рисунок 3.13 – Результат моделі SVM з RSI

```
> print(model3_metrics)
 Accuracy Precision    Recall  F1 Score
0.7768116 0.7685185 0.8601036 0.8117359
```

Рисунок 3.14 – Результат моделі SVM без RSI

На рисунках 3.13-3.14 представлено два набори результатів оцінки моделі машини опорних векторів (SVM) для задачі класифікації.

Перша модель, що включає індекс відносної сили (RSI) як один з предикторів, показала наступні результати: точність (Accuracy) складає 0.7913, влучність (Precision) — 0.7915, повнота (Recall) — 0.8445 та F1-оцінка — 0.8190. Ці показники свідчать про високу ефективність моделі у правильному класифікуванні випадків та баланс між виявленням позитивних класів і уникненням помилок.

Друга модель SVM, яка не включає RSI в якості предиктора, показує дещо інші характеристики: точність має значення 0.7768, влучність — 0.7685, повнота — 0.8601, а F1-оцінка знижується до 0.8117. Незважаючи на високу повноту, зниження точності та F1-оцінки може вказувати на те, що модель без RSI більше схильна до помилково позитивних класифікацій, що призводить до меншої загальної точності моделі.

3.3 Оцінка моделей

Для порівняння моделей були використані наступні метрики:

– accuracy – частка правильно класифікованих екземплярів щодо загальної кількості екземплярів;

– precision – частка чітко класифікованих позитивних екземплярів щодо всіх позитивних прогнозів;

– recall – частка правильно класифікованих позитивних екземплярів щодо всіх дійсних позитивних екземплярів;

– f1 score – гармонічне середнє влучності та повноти.

Таблиця 3.1– Фінальні метрики моделей

	Accuracy	Precision	Recall	F1 Score
логістична регресія	0.7768116	0.7636364	0.8704663	0.8135593
KNN	0.7681159	0.7627907	0.8497409	0.8039216
SVM	0.7768116	0.7685185	0.8601036	0.8117359
логістична регресія +RSI	0.7855072	0.7874396	0.8445596	0.8150000
KNN+RSI	0.7942029	0.8144330	0.8186528	0.8165375
SVM+RSI	0.7913043	0.7951220	0.8445596	0.8190955

ROC-крива є графічним представленням якості бінарних класифікаторів при різних порогах дискримінації. Вона показує залежність між чутливістю (Sensitivity або True Positive Rate) та специфічністю (1 - Specificity або False Positive Rate).

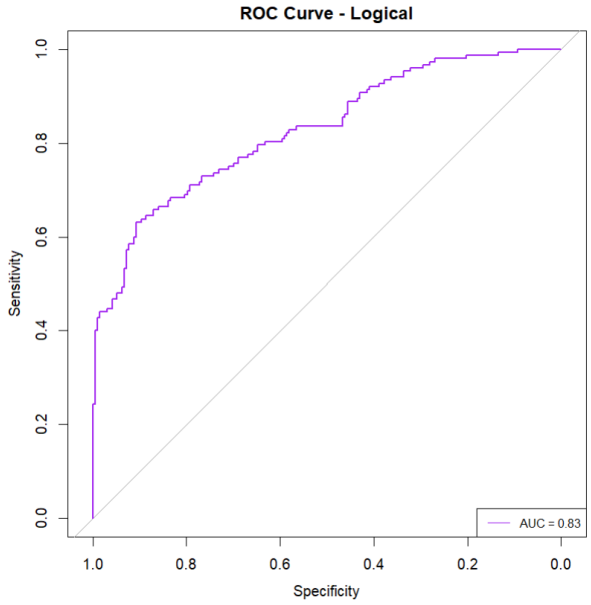

Рисунок 3.15 – ROC Логістичної регресії

Згідно рисунку 3.15, AUC для логістичної регресії становить 0.83, що є дуже хорошим результатом і вказує на високу здатність моделі розрізняти між позитивними та негативними класами. Це значення свідчить про те, що модель має сильні прогностичні властивості.

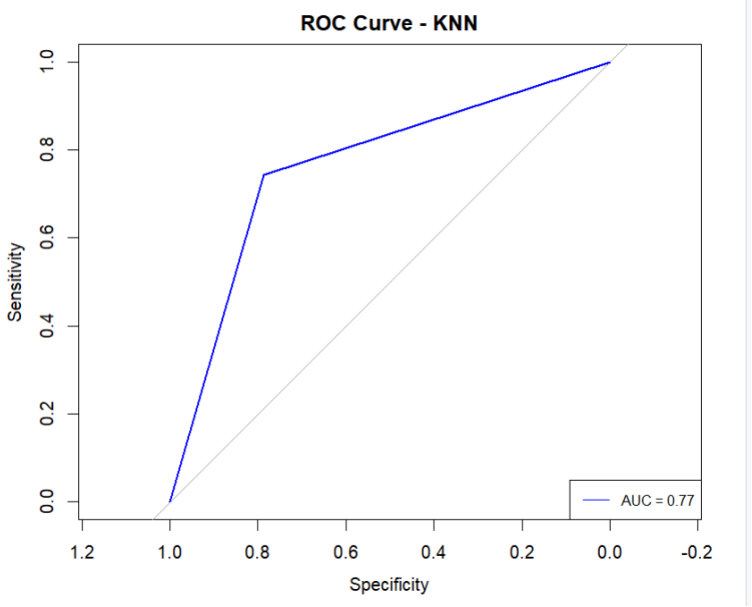

Рисунок 3.16 – ROC моделі k-найближчих сусідів

Значення AUC 0.77 для моделі KNN свідчить про те, що модель має добру здатність відрізняти між позитивними та негативними класами, але є місце для покращення, оскільки вона не досягає ідеального показника.

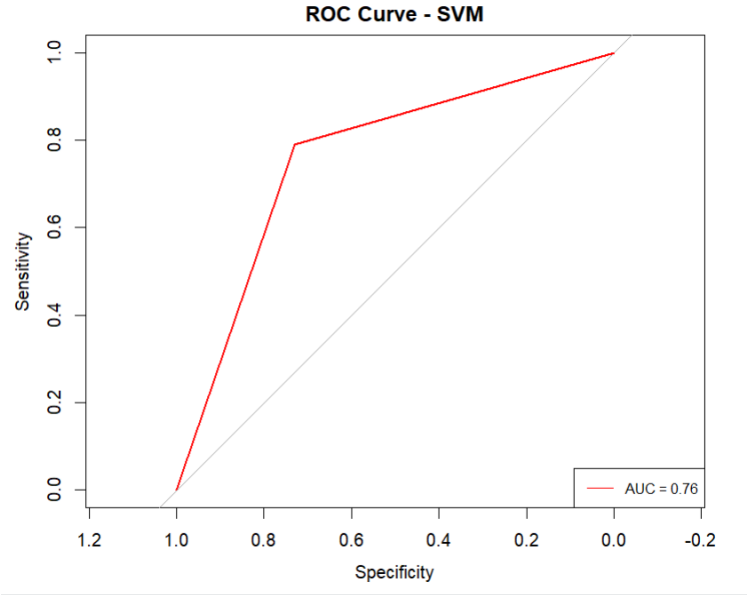

Рисунок 3.17 – ROC SVM

Для моделі SVM, показник AUC становить 0.76, що свідчить про досить високу здатність класифікатора розрізняти між позитивними та негативними класами. Хоча цей показник і не є ідеальним, він все ж вищий за середній рівень і вказує на ефективність моделі у визначенні випадків, які належать до різних класів.

Висновоки до розділу 3

На основі результатів класифікаційних моделей (логістична регресія, K-Nearest Neighbors - KNN і Support Vector Machine - SVM) можна зробити наступні узагальнені висновки.

Логістична регресія без RSI продемонструвала точність (Accuracy) 0.7768, влучність (Precision) 0.7636, повнота (Recall) 0.8705, і F1-оцінку 0.8136. З RSI ці показники стали: точність 0.7855, влучність 0.7874, повнота 0.8446, і F1-оцінка 0.8150. Це свідчить про те, що RSI допоміг покращити точність і F1-оцінку, але знизив відклик.

Модель KNN без RSI мала точність 0.7681, влучність 0.7628, повнота 0.8497, і F1-оцінку 0.8039. При використанні RSI показники стали: точність 0.7942, влучність 0.8144, повнота 0.8187, і F1-оцінка 0.8165. Таким чином, RSI суттєво підвищив точність і влучність, одночасно знижуючи відклик.

SVM без RSI показала точність 0.7768, влучність 0.7685, повнота 0.8601, і F1-оцінку 0.8117. З включенням RSI ці показники змінилися до: точність 0.7913, влучність 0.7951, повнота 0.8446, і F1-оцінка 0.8191. В результаті, RSI збільшив точність і F1-оцінку, водночас злегка знижуючи відклик.

Узагальнюючи, додавання RSI як предиктора у всіх трьох моделях призвело до покращення точності та F1-оцінки, що свідчить про його значення як корисного інструменту для підвищення загальної ефективності моделей у задачах класифікації. Водночас, можна відзначити зниження відклику після включення RSI, що може бути важливим для розробки стратегій, де важливіше зменшити кількість пропущених позитивних випадків.

ВИСНОВКИ

У кваліфікаційній роботі магістра зосереджено увагу на важливій задачі класифікації напрямку росту активів, актуальній у сучасних умовах динамічного фінансового ринку. Робота охоплює комплексний аналіз сучасного стану цієї проблематики, включаючи теоретичні основи, огляд останніх досліджень та публікацій, а також детальну постановку задачі з урахуванням специфіки ринку активів.

У другому розділі розглянуто різні технології та інструменти, що можуть бути застосовані для вирішення поставленої задачі. Представлено детальний огляд моделей для класифікації, особливостей мови програмування R, її середовища розробки RStudio, а також методів технічного аналізу. Окрему увагу приділено технологіям, що надають додаткові дані для аналізу, таким як Santiment, та описано метрики, які використовуються для оцінки ефективності моделей.

Третій розділ присвячено програмній реалізації регресивних моделей. Він охоплює підготовку даних, створення моделей логістичної регресії, KNN та SVM з використанням та без використання RSI як предиктора, а також оцінку їх ефективності за допомогою різних метрик. Основними висновками цього розділу є те, що включення RSI як предиктора покращує точність і F1-оцінку моделей, хоча і знижує повноту.

Загальний висновок кваліфікаційної роботи підкреслює, що використання різноманітних методів технічного аналізу та машинного навчання може значно покращити здатність прогнозування напрямку росту активів. Виявлено, що RSI є корисним інструментом у багатьох моделях, а мова програмування R разом з середовищем RStudio надає потужні і гнучкі засоби для аналізу та моделювання даних. Також, враховуючи швидкі зміни на фінансових ринках, аналіз настроїв може бути використаний для підвищення точності розуміння напрямку цін на активи, що надасть інвесторам та аналітикам важливі інструменти для прийняття обґрунтованих інвестиційних рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Predicting Stock Price Movements Using Machine Learning: A Sentiment Analysis Approach URL: https://www.researchgate.net/publication/364609314_Predicting_Stock_Price_Movements_Using_Machine_Learning_A_Sentiment_Analysis_Approach (дата звернення: 16.12.2023)

2. Patel, Jigar, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. 2015. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417414004473?via%3Dihub (дата звернення: 17.12.2023)

3. Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine URL: https://ieeexplore.ieee.org/document/8326522 (дата звернення: 17.12.2023)

4. Do negative events really have deteriorating effects on stock performance? A comparative study on Tesla (US) and Nio (China) URL: https://www.emerald.com/insight/content/doi/10.1108/JABES-07-2021-0106/full/html (дата звернення: 17.12.2023)

5. Nabipour, Mojtaba, Pooyan Nayyeri, Hamed Jabani, S. Shahab, and Amir Mosavi. 2020. Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis. URL: https://ieeexplore.ieee.org/abstract/document/9165760/

6. Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implicationshttps URL: https://www.mdpi.com/2227-7072/11/3/94 (дата звернення: 17.12.2023)

7. Efficient Capital Markets: A Review of Theory and Empirical Work URL : https://www.jstor.org/stable/2325486 (дата звернення: 17.12.2023)

8. Extreme sentiment and herding: Evidence from the cryptocurrency market URL: https://www.sciencedirect.com/science/article/abs/pii/S0275531922001568 (дата звернення: 17.12.2023)

9. Sentiments Extracted from News and Stock Market Reactions in Vietnam URL: https://www.mdpi.com/2227-7072/11/3/101 (дата звернення: 17.12.2023)

10. Twitter mood predicts the stock market URL: https://www.sciencedirect.com/science/article/abs/pii/S187775031100007X (дата звернення: 17.12.2023)

11. Tweets and Trades: the Information Content of Stock Microblogs URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-036X.2013.12007.x (дата звернення: 17.12.2023)

12. Predicting Stock Price Movement Using Sentiment Analysis and CandleStick Chart Representation URL : https://www.ajssmt.com/Papers/530118.pdf (дата звернення: 17.12.2023)

13. The Fed and the stock market: A tale of sentiment states URL: https://www.sciencedirect.com/science/article/pii/S0261560622001103 (дата звернення: 17.12.2023)

14. Impact of Liquidity and Investors Sentiment on Herd Behavior in Cryptocurrency Market URL: https://www.mdpi.com/2227-7072/11/3/97 (дата звернення: 17.12.2023)

15. Prädiktion von Aktienkursen mit Neuronalen Netzen URL: https://isl.anthropomatik.kit.edu/pdf/Handloser2017.pdf (дата звернення: 17.12.2023)

16. INVESTOR SENTIMENT AND THE CROSS-SECTION OF STOCK RETURNS URL: https://www.nber.org/system/files/working_papers/w10449/w10449.pdf (дата звернення: 17.12.2023)

17. Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong URL: https://www.sciencedirect.com/science/article/abs/pii/S0306457319307952?via%3Dihub (дата звернення: 17.12.2023)

18. Unusual News Flow and the Cross Section of Stock Returns URL: https://escholarship.org/content/qt0fp7n83b/qt0fp7n83b.pdf?t=old1ff (дата звернення: 17.12.2023)

19. Investor sentiment and the near-term stock market URL: https://linkinghub.elsevier.com/retrieve/pii/S0927539803000422 (дата звернення: 18.12.2023)

20. Herding and positive feedback trading on property stocks URL: https://www.emerald.com/insight/content/doi/10.1108/14635780810857872/full/html (дата звернення: 18.12.2023)

21. ESG performance, herding behavior and stock market returns: evidence from Europe URL: https://link.springer.com/article/10.1007/s12351-023-00745-1 (дата звернення: 18.12.2023)

22. Does Russia–Ukraine war generate herding behavior in Moscow Exchange? URL: https://www.emerald.com/insight/content/doi/10.1108/RBF-01-2023-0014/full/html (дата звернення: 18.12.2023)

23. Do investors herd in a volatile market? Evidence of dynamic herding in Taiwan, China, and US stock markets URL: https://www.sciencedirect.com/science/article/abs/pii/S1544612322005414 (дата звернення: 18.12.2023)

24. Herding Behavior in Developed, Emerging, and Frontier European Stock Markets during COVID-19 Pandemic URL: https://www.mdpi.com/1911-8074/15/9/400 (дата звернення: 18.12.2023)

25. Do investors herd in a volatile market? Evidence of dynamic herding in Taiwan, China, and US stock markets URL: https://linkinghub.elsevier.com/retrieve/pii/S1544612322005414 (дата звернення: 18.12.2023)

26. Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications URL: https://www.mdpi.com/2227-7072/11/3/94 (дата звернення: 18.12.2023)

27. Prediction of stock market using Artificial Intelligence URL: https://deliverypdf.ssrn.com/delivery.php?ID=339119017068084104073007094121106006099039071063064018087113119025127090088101003098102118004001052027117106104124005000122072109094092045027121066115088065126126087070047024085093009115126124088125077071090027125116026091019108108008092103099089126087&EXT=pdf&INDEX=TRUE (дата звернення: 18.12.2023)

28. What is Logistic regression? URL: https://www.ibm.com/topics/logistic-regression (дата звернення: 17.12.2023)

29. Logistic Regression URL: https://web.stanford.edu/~jurafsky/slp3/5.pdf (дата звернення: 17.12.2023)

30. Logistic Regression in Machine Learning URL: https://www.geeksforgeeks.org/understanding-logistic-regression/ (дата звернення: 17.12.2023)

31. K-najbliższych sąsiadów URL: https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstknn.html (дата звернення: 22.01.2024)

32. A Complete Guide to K-Nearest Neighbors (Updated 2024) URL: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/ (дата звернення: 22.01.2024)

33. K-Nearest Neighbors (KNN) Classification with scikit-learn URL: https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn (дата звернення: 22.01.2024)

34. K-Nearest Neighbor (KNN) Algorithm for Machine Learning URL: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning (дата звернення: 22.01.2024)

35. Metoda wektorów nośnych URL: https://www.statsoft.pl/textbook/stathome_stat.html?https%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstsvm.html (дата звернення: 22.01.2024)

36. Метод опорних векторів SVM URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0

%B4_%D0%BE%D0%BF%D0%BE%D1%80%D0%BD%D1%8B%D1%85_%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BE%D0%B2_(SVM) (дата звернення: 22.01.2024)

37. SVM – Support Vector Machines Metoda wektorów no śnych URL: https://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf (дата звернення: 22.01.2024)

38. Guide on Support Vector Machine (SVM) Algorithm URL: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/ (дата звернення: 22.01.2024)

39. Проект R для статистичних обчислень. URL: https://www.r-project.org (дата звернення: 22.01.2024)

40. ЗАСТОСУВАННЯ ПРОГРАМНОГО ПАКЕТУ R URL: https://www.researchgate.net/publication/331401481_ZASTOSUVANNA_PROGRAMNOGO_PAKETU_R_U_NAUKOVIH_DOSLIDZENNAH_MAJBUTNIH_FILOLOGIV (дата звернення: 22.01.2024)

41. The Comprehensive R Archive Network URL: https://cran.r-project.org/ (дата звернення: 22.01.2024)

42. DataCamp: Learn Data Science and AI Online URL: https://www.datacamp.com/projects/177 (дата звернення: 22.01.2024)

43. Joe Cheng - Interview by DataScience.LA at useR 2014 URL: https://www.youtube.com/watch?v=uJm-its3ZWM (дата звернення: 22.01.2024)

44. Найбільш надійна IDE для науки про дані з відкритим кодом RStudio URL: https://posit.co/ (дата звернення: 22.01.2024)

45. Технічний аналіз – посібник для початківців URL: https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/technical-analysis (дата звернення: 21.01.2024)

46. Relative Strength Index (RSI) Indicator Explained With Formula URL: https://www.investopedia.com/terms/r/rsi.asp (дата звернення: 23.01.2024)

47. Santiment crypto intelligence tools URL: https://app.santiment.net/ (дата звернення: 23.01.2024)

48. Sanbase URL: https://play.google.com/store/apps/details?id=net.santiment.sanbase.android&hl=pl (дата звернення: 23.01.2024)

49. Crypto News API URL: https://cryptonews-api.com/ (дата звернення: 18.12.2023)

ДОДАТОК А

Код програмної реалізації

```
# Встановлення та завантаження бібліотек
install.packages(c("httr", "jsonlite"))
install.packages("quantmod")
install.packages("caret")
install.packages("pROC")
install.packages("ROCR")
install.packages("class")
install.packages("e1071")
library(e1071)
library(class)
library(httr)
library(jsonlite)
library(quantmod)
library(caret)
library(pROC)
library(ROCR)
library(dplyr)

request <- function(page){
  # Виконання HTTP-запиту
  url<-paste0("https://cryptonews-api.com/api/v1/stat?&tickers=BTC&date=01152019-
02082024&page=",page,"&token=secret")
  response <- GET(url)
  # Перевірка успішності запиту
  if (http_type(response) == "application/json") {
    # Парсинг JSON
    json_content <- content(response, "text", encoding = "UTF-8")
    parsed_data <- fromJSON(json_content)
    return(parsed_data)
  } else {
    warning("Ошибка запроса")
  }
}
parsed_data <-request(1)
```

```
news <- data.frame(Date = as.Date(character()), Neutral = double(), Positive = double(), Negative  = double())

# Перетворення векторів даних на датафрейм
i<-1
while(i < parsed_data$total_pages){
  for (dateN in names(parsed_data$data)) {
    date <- parsed_data$data[[dateN]]$BTC
    sum <-date$Neutral+date$Positive+date$Negative
    temp <- data.frame(
      Date = as.Date(dateN),
      Neutral = date$Neutral/sum,
      Positive = date$Positive/sum,
      Negative = date$Negative/sum
    )
    news <- rbind(news, temp)
  }
  i<-i+1
  parsed_data<-request(i)
}
write.csv(news, file = "news_data.csv", row.names = FALSE)

# Завантаження даних з біткоїну з Yahoo Finance
getSymbols("BTC-USD", src = "yahoo", from = "2020-11-19", to = Sys.Date())
# Створення змінної зростання/падіння ціни
price_changes <-diff(Cl(`BTC-USD`)) > 0

#Замість API
news <- read.csv(file = "news_data.csv")
news$Date <- as.Date(news$Date, format="%Y-%m-%d")

price_changes_df <- data.frame(Date = index(price_changes), dPrice = price_changes$`BTC-USD.Close`)

# Підрахунок RSI
s<-RSI(`BTC-USD`$`BTC-USD.Close`)
s$rsi<-s$rsi/100
rsi <- data.frame(Date = index(s), rsi = s$rsi)
```

```
rsi <- rsi %>%
  slice((14 + 1):n())
rsiData <- merge(news, rsi, by = "Date", all.x = TRUE)

# Старт
merged_data <- merge(news, price_changes_df, by = "Date", all.x = TRUE)
merged_data <- merge(merged_data, rsi, by = "Date", all.x = TRUE)

colnames(merged_data)[colnames(merged_data) == "BTC.USD.Close"] <- "dPrice"
merged_data$dPrice <- as.numeric(merged_data$dPrice)



set.seed(123)  # для воспроизводимости
evaluate_model <- function(true_values, predicted_values) {
  conf_matrix           <-           confusionMatrix(as.factor(predicted_values),
as.factor(true_values))
  accuracy <- conf_matrix$overall[['Accuracy']]
  precision <- conf_matrix$byClass[['Pos Pred Value']]
  recall <- conf_matrix$byClass[['Sensitivity']]
  f1_score <- conf_matrix$byClass[["F1"]]

  return(c(Accuracy=accuracy,     Precision=precision,     Recall=recall,     `F1
Score`=f1_score))
}

# Разделение данных на обучающую и тестовую выборки
set.seed(123)  # для воспроизводимости
sample_index <- sample(1:nrow(merged_data), 0.7 * nrow(merged_data))  # 70% данных
в обучающей выборке
train_data <- merged_data[sample_index, ]
test_data <- merged_data[-sample_index, ]
test_data$dPrice <- as.factor(test_data$dPrice)

# Створення вектора з назвами предикторів
predictors <- c("Positive", "Negative", "Neutral"
                #,"rsi"
               )
```

```
# Використання вектора для створення формули
formula <- as.formula(paste("dPrice ~", paste(predictors, collapse = " + ")))



###LOGICAL
# Обучение модели линейной регрессии
model <- glm(formula, data = train_data, family = "binomial")

# Предсказание на тестовой выборке
predictions <- predict(model, newdata = test_data)
predicted_classes <- ifelse(predictions > 0.5, 1, 0)

#оцінка
model1_metrics<-evaluate_model(test_data$dPrice,predicted_classes)
print(model1_metrics)
###KNN
# Обучение KNN модели
set.seed(123)
k_values <- 1:400
accuracy_scores <- numeric(length(k_values))
for (i in 1:length(k_values)) {
  k <- k_values[i]

  # Навчіть модель KNN з поточним значенням k
  knn_modelT <- knn(train = train_data[, predictors],
                    test = test_data[, predictors],
                    cl = train_data$dPrice,
                    k = k)

  # Оцініть точність на валідаційній вибірці і збережіть її
  accuracy_scores[i] <- mean(knn_modelT == test_data$dPrice)
}

# Знайдіть оптимальне значення k
best_k <- k_values[which.max(accuracy_scores)]
cat("Найкраще значення k:", best_k, "\n")

knn_model <- knn(train = train_data[, predictors],
```

```
                test = test_data[, predictors],
                cl = train_data$dPrice,
                k = 4)
# Оцінка моделі
model2_metrics<-evaluate_model(test_data$dPrice,knn_model)
print(model2_metrics)



###SVM
# Обучение SVM модели
svm_model <- svm(formula, data = train_data)

# Предсказание на тестовой выборке
svm_pred <- predict(svm_model, newdata = test_data[, predictors])

# Підготовка до оцінки
svm_predicted_classes <- ifelse(svm_pred > 0.5, 1, 0)

# Оцінка моделі
model3_metrics = evaluate_model(test_data$dPrice,svm_predicted_classes)
print(model3_metrics)

results_df <- data.frame(rbind(model1_metrics, model2_metrics, model3_metrics))
rownames(results_df) <- c("Logical", "KNN", "SVM")
colnames(results_df) <- c("Accuracy","Precision","Recall","F1")
print(results_df)



# ROC-крива и AUC-ROC

coefficients <- coef(model)
print(coefficients)

roc_curve <- roc(test_data$dPrice, predictions)
print(paste("AUC-ROC:", auc(roc_curve)))

plot(roc_curve, main = "ROC Curve - Logical", col = "purple", lwd = 2)
```

```
legend("bottomright", legend = paste("AUC =", round(auc(roc_curve), 2)), col = "purple", lty = 1, cex = 0.8)

roc_knn <- roc(test_data$dPrice, as.numeric(knn_model))
auc_knn <- auc(roc_knn)
plot(roc_knn, main = "ROC Curve - KNN", col = "blue", lwd = 2)
legend("bottomright", legend = paste("AUC =", round(auc_knn, 2)), col = "blue", lty = 1, cex = 0.8)

roc_svm <- roc(test_data$dPrice, as.numeric(as.factor(predicted_classes)))
auc_svm <- auc(roc_svm)
plot(roc_svm, main = "ROC Curve - SVM", col = "red", lwd = 2)
legend("bottomright", legend = paste("AUC =", round(auc_svm, 2)), col = "red", lty = 1, cex = 0.8)
```