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
«____»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЛАСИФІКАЦІЇ ВОЛАТИЛЬНОСТІ ВАЛЮТ

Спеціальність 122 «Комп'ютерні науки»

122 – КРМ – 601.21810310

Виконала студентка 6-го курсу, групи 601
_____ ***А. А. Єськіна***
(підпис, ініціали та прізвище)
«___» _____ 2024 р.

Керівник: канд. фіз.-мат. наук, доцент
_____***І. В. Кулаковська***
(підпис, ініціали та прізвище)
«___» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

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

Рівень вищої освіти **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
«____» _____2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Єськіній Анні Андріївні
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи магістра «Інтелектуальна система класифікації волатильності валют».

Керівник роботи Кулаковська І. В., канд. фіз.-мат. наук, доцент.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затв. наказом Ректора ЧНУ ім. Петра Могили від «__» _____ 202_ р. № ___

2. Строк подання студентом роботи «19» _лютого_ 2024 р.

3. Вхідні (початкові) дані до роботи: поняття волатильності валют, загальні відомості машинного навчання, методів класифікації та інтелектуальних систем.

Очікуваний результат роботи: реалізація інтелектуальної системи класифікації волатильності валют на основі обраних методів класифікації.

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):

- аналіз актуальності обраної теми;

- розгляд та вивчення загальної теорії;
- обґрунтування засобів програмної реалізації;
- процес моделювання та програмна реалізація інтелектуальної системи класифікації волатильності валют;
- тестування функціональності телеграм-бота та отримання результатів роботи.

5. Перелік графічних матеріалів 3 таблиці, 60 рисунків, презентація.

6. Завдання до спеціальної частини: «Забезпечення вимог охорони праці у приміщенні комп'ютерної лабораторії вищого навчального закладу».

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. фіз.-мат. наук, доцент Кулаковська І. В.**	

Керівник роботи канд. фіз.-мат. наук, доцент Кулаковська І. В._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання __Єськіна А. А.__
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «_31_» _жовтня_ 2023 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи магістра

Тема: Інтелектуальна система класифікації волатильності валют_____

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми КРМ. Подання заяви на затвердження теми КРМ	01.09.2023	10.10.2023	Виконано
2	Отримання завдання на виконання КРМ	11.10.2023	01.11.2023	Виконано
3	Складання календарного плану роботи на період виконання КРМ	02.11.2023	10.11.2023	Виконано
4	Проходження передатестаційної практики, збір та аналіз матеріалів до КРМ	27.11.2023	23.12.2023	Виконано
5	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	25.12.2023	12.01.2024	Виконано
6	Опис фахової частини КРМ, аналіз тенденцій обраної теми, огляд методів класифікації, програмна реалізація інтелектуальної системи	13.01.2024	25.01.2024	Виконано
7	Розробка спеціальної частини з охорони праці та методичної частини	25.01.2024	28.01.2024	Виконано
8	Перший попередній захист КРМ на засіданні комісії кафедри	29.01.2024	29.01.2024	Виконано
9	Корегування роботи за результатами попереднього захисту	30.01.2024	05.02.2024	Виконано
10	Доробка та остаточне оформлення КРМ	06.02.2024	11.02.2024	Виконано
11	Другий попередній захист КРМ на засіданні комісії кафедри	12.02.2024	12.02.2024	Виконано
12	Подання КРМ, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.02.2024	20.02.2024	Виконано
13	Захист КРМ перед екзаменаційною комісією (ЕК)	26.02.2024	27.02.2024	Виконано

Розробив студент __Єськіна А. А._____ _____
(прізвище, ім'я, по батькові студента) (*підпис*)

Керівник роботи __канд. фіз.-мат. наук, доцент Кулаковська І. В._ _____
(*посада, прізвище, ім'я, по батькові)* *(підпис)*

«09» листопада 2023 р.

АНОТАЦІЯ

до кваліфікаційної роботи магістра
студентки групи 601 ЧНУ ім. Петра Могили

Єськіної Анни Андріївни

на тему: **“ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЛАСИФІКАЦІЇ ВОЛАТИЛЬНОСТІ ВАЛЮТ”**

Актуальність даного дослідження полягає у необхідності підвищення якості класифікації волатильності валют за рахунок методів машинного навчання та аналізу динаміки валютних ринків. Таким чином трейдери та аналітики матимуть змогу отримати більш точні та передбачувані результати для прийняття ефективних фінансових рішень, а також краще розуміти ризики та можливості на валютному ринку. Вони зможуть вчасно реагувати на зміни у валютному середовищі та вдосконалювати свої стратегії торгівлі.

Особливо, з урахуванням нестабільності сучасного економічного середовища та впливу різноманітних факторів, таких як економічні новини, глобальні кризи, та природні катастрофи, розробка інтелектуальної системи для класифікації волатильності валют стає критично важливою.

Об’єктом дослідження є процес класифікації та аналізу даних щодо волатильності валют використовуючи методи машинного навчання.

Предметом дослідження є моделі та методи машинного навчання, а саме наївний баєсівський класифікатор та метод опорних векторів для вирішення задач класифікації на основі сформованого набору даних.

Метою дослідження є розробка та дослідження інтелектуальної системи, яка здатна ефективно класифікувати рівень волатильності на валютних ринках. Робота спрямована на вивчення різних методів машинного навчання та класифікаторів на основі різних мір подібності, з метою визначення оптимального підходу для класифікації волатильності

Дана робота складається з п’яти розділів. Кожен розділ відповідно присвячений: аналізу предметної області, методам машинного навчання, використаним у кваліфікаційній роботі магістра, моделюванню і розробці системи, аналізу отриманих результатів, охороні праці та методичній частині магістерської роботи. Загальний обсяг роботи – 129 сторінок. Кваліфікаційна робота магістра містить один додаток, 9 формул, 60 рисунків, 3 таблиці і посилання на 41 літературне джерело.

Ключові слова: інтелектуальна система, методи машинного навчання, класифікація, метрики оцінки якості класифікації, волатильність.

ABSTRACT

to the master's qualification work by the student of the group 601 of Petro Mohyla Black Sea National University

Yeskina Anna

“INTELLIGENT SYSTEM FOR CURRENCY VOLATILITY CLASSIFICATION”

The relevance of this study lies in the need to improve the quality of currency volatility classification using machine learning methods and analysis of currency market dynamics. In this way, traders and analysts will be able to obtain more accurate and predictable results for making effective financial decisions, as well as better understand the risks and opportunities in the foreign exchange market. They will be able to respond to changes in the currency environment in a timely manner and improve their trading strategies.

Especially given the instability of the current economic environment and the impact of various factors such as economic news, global crises, and natural disasters, the development of an intelligent system for classifying currency volatility becomes critical.

The object of research is the process of classifying and analyzing data on currency volatility using machine learning methods.

The subject of the study is machine learning models and methods, namely the naive Bayesian classifier and the support vector method for solving classification problems based on the generated data set.

The aim of the study is to develop and research an intelligent system that can effectively classify the level of volatility in the currency markets. The work is aimed at studying various machine learning methods and classifiers based on different similarity measures in order to determine the optimal approach for volatility classification.

This thesis consists of five chapters. Each chapter is devoted to the analysis of the subject area, machine learning methods used in the master's thesis, modeling and development of the system, analysis of the results, labor protection, and the methodological

part of the master's thesis. The total volume of the work is 129 pages. The master's qualification work contains one appendix, 9 formulas, 60 figures, 3 tables and references to 41 literature sources.

Keywords: intelligent system, machine learning methods, classification, classification quality assessment metrics, volatility.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ЗМІ – засоби масової інформації

МН – машинне навчання

СМОП – система миттєвого обміну повідомленнями

ФРС – Федеральна резервна система

API – Application Programming Interface

ATR – Average True Range

CSV – Comma-Separated Values

E2E – End-to-end

JSON – JavaScript Object Notation

NLP – Natural Language Processing

SVM – Support Vector Machines

TON – Telegram Open Network

XML – eXtensible Markup Language

IM – Instant messaging

ICO – Initial Coin Offering

ВСТУП

Сучасний світ неможливо уявити без наступного елемента – фінансів. Фінанси стали ниткою, що сплітає різні аспекти нашого життя, від покупок продуктів і оплати комунальних послуг до інвестування для майбутнього та планування власного бюджету. Виправно, фінансова галузь є не лише основою для функціонування господарства, але й однією з найбільш динамічних та технологічно розвинених галузей. Вона зазнає постійних змін та вдосконалень завдяки технологічному прогресу і потребам сучасного суспільства.

Фінанси відіграють важливу роль у різноманітних сферах нашого життя, від особистих фінансів кожного з нас до глобальних економічних процесів. Ця сфера вимагає постійного аналізу, прогнозування та прийняття рішень в реальному часі. Водночас, розвиток сучасних технологій, таких як штучний інтелект і машинне навчання, відкриває нові можливості для поліпшення аналізу та прийняття рішень у фінансовій галузі.

Крім цього, ураховуючи зростання ролі соціальних мереж у сучасному інформаційному суспільстві, актуальним є вивчення можливостей та обмежень використання цих платформ для класифікації ринкових тенденцій. Визначення точності та надійності цього підходу вимагає глибокого аналізу обраної платформи, а саме – Телеграму.

Актуальність кваліфікаційної роботи магістра обумовлена важливістю вивчення та аналізу динаміки валютних ринків. З урахуванням нестабільності сучасного економічного середовища та впливу різноманітних факторів, таких як економічні новини, глобальні кризи, та природні катастрофи, розробка інтелектуальної системи для класифікації волатильності валют стає критично важливою.

Така система може використовуватися для прогнозування та управління ризиками на валютних ринках, що є актуальним завданням для фінансових аналітиків, трейдерів та інших учасників фінансового ринку. Враховуючи постійні зміни у світовій економіці, стабільні та ефективні методи класифікації волатильності допоможуть у забезпеченні успішних стратегій торгівлі та прийнятті обґрунтованих рішень на фінансових ринках.

Об'єкт дослідження – процес класифікації волатильності валют, використовуючи методи машинного навчання.

Предмет дослідження – моделі та методи машинного навчання, а саме наївний баєсівський класифікатор та метод опорних векторів для задачі класифікації.

Наприклад, дослідник може вивчати, як кожен з цих методів може адаптуватися до особливостей динаміки валютних ринків та які з них є більш ефективними у прогнозуванні волатильності. Порівнюючи їхні результати та враховуючи різні міри подібності, дослідник ставить за мету визначити оптимальний підхід для класифікації валютної волатильності.

Метою роботи є розробка та дослідження інтелектуальної системи, яка здатна ефективно класифікувати рівень волатильності на валютних ринках. Робота спрямована на вивчення різних методів машинного навчання та класифікаторів на основі різних мір подібності, з метою визначення оптимального підходу для класифікації волатильності.

Окрім того, робота спрямована на аналіз та порівняння результатів різних методів, враховуючи різноманітні аспекти фінансових ринків. Очікується, що розроблена інтелектуальна система стане інструментом для управління ризиками та прийняття рішень на валютних ринках, допомагаючи трейдерам та інвесторам ефективно адаптуватися до змін у рівнях волатильності.

Апробація результатів дослідження. Основні положення кваліфікаційної роботи магістра, зокрема інтелектуальна система класифікації волатильності валют,

викладено в матеріалах XXVI Всеукраїнської науково-практичної конференції «Могилянські читання-2023. Досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти» та Всеукраїнській науково-практичній конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія».

Структура роботи. Магістерська кваліфікаційна робота складається зі вступу, 3 розділів, висновків та списку використаних джерел, що нараховує 41 позицію, 60 рисунків, 9 формул, 3 таблиці та одного додатку. Загальний обсяг 129 сторінок.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Технологічні та методологічні тенденції у розробці інтелектуальних систем

Сучасний фінансовий ринок постійно змінюється, і ефективний аналіз волатильності валют є критичним для успішної стратегії управління ризиками та інвестування. У цьому контексті виникає необхідність у нових підходах до класифікації та прогнозування, інтегруючи інтелектуальні системи та технологію чат-ботів [2].

Чат-бот – це програмний агент, створений для автоматизованої взаємодії з користувачами за допомогою тексту або голосових повідомлень. Він може використовувати різні технології, такі як штучний інтелект, обробка природної мови та автоматизовані алгоритми, для розуміння контексту розмови та генерування інформативних відповідей на запитання користувачів. Вони були створені в 1960-х роках, але їхня технологія продовжує розвиватися.

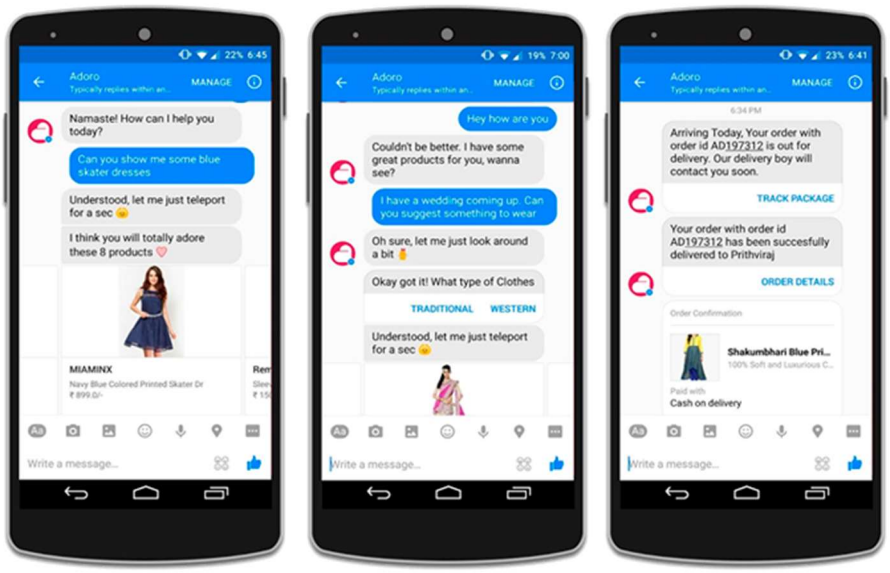


Рисунок 1.1 – Демонстрація чат-боту

Одним з перших та найвідоміших чат-ботів є ELIZA, який був розроблений у 1966 році. ELIZA імітував розмову з психотерапевтом, використовуючи набори ключових слів і відповідей.

Рисунок 1.2 – Огляд чат-боту ELIZA

За останні десятиріччя методи машинного навчання (МН) значно трансформували підхід до аналізу фінансового ринку. Інноваційні технології дозволяють ефективно використовувати великі обсяги даних для класифікації ринкових тенденцій, ризиків та оптимізації інвестиційного портфеля.

Розвиток технологій NLP дозволяє інтелектуальним системам ефективно взаємодіяти з користувачами через природну мову. Це стає важливим аспектом для чат-ботів, які використовуються в прогнозуванні фінансового ринку через платформу Telegram.

У традиційному NLP слова розглядаються як дискретні одиниці, які представляються у вигляді одновимірних векторів. Такий спосіб представлення має недолік: він не дозволяє визначити схожість між словами.

Важливо зазначити, що не існує універсального методу, який підходить до всіх ситуацій, і використання машинного навчання вимагає постійного вдосконалення та адаптації до змін на фінансових ринках. Наслідком правильного використання може бути покращення стратегій та досягнення кращих фінансових результатів.

Інтелектуальні системи класифікації волатильності валют використовують алгоритми машинного навчання та штучного інтелекту для обробки великого обсягу фінансових даних. Ці системи можуть автоматично аналізувати патерни та визначати фактори, що впливають на волатильність різних валют.

У свою чергу, використання чат-ботів, зокрема на платформі Telegram, дозволяє здійснювати швидкий та зручний доступ до результатів аналізу в реальному часі. Інвестори та фахівці можуть отримувати повідомлення та консультації щодо прогнозів безпосередньо через чат-інтерфейс.

Такий інтегрований підхід дозволяє не лише ефективно аналізувати волатильність валют, але і реагувати на зміни ринку в реальному часі, забезпечуючи більш точне управління портфелем та зниження ризиків.

У різноманітному світі фінансів та інтелектуальних систем через Telegram-бот вже існують низка вразливих, але успішних реалізацій.

Однією з таких платформ є "StockBot," що використовується для моніторингу та аналізу фінансових ринків. Його успішний аспект полягає в здатності надавати реальний час та персоналізовану інформацію щодо акцій, курсів валют та інших фінансових інструментів. Користувачі можуть отримати швидкий та точний аналіз, що полегшує прийняття фінансових рішень [9].

Іншим прикладом є "CryptoAnalyzer", який зосереджений на криптовалютному аналізі. Цей бот взаємодіє з користувачами через Telegram, надаючи їм зручний

доступ до аналізу ринку криптовалют, прогнозів цін та загальної динаміки [5]. Високоризикований аспект полягає у залежності від волатильності крипторинку, але успішний бот вигідно використовує цю волатильність для забезпечення користувачам актуальної та вигідної інформації.

Ось деякі з основних можливостей та функціоналу "CryptoAnalyzer" бота:

1) аналіз волатильності: бот здатен автоматично аналізувати волатильність різних криптовалютних пар. Він використовує алгоритми машинного навчання для виявлення патернів та прогнозування рівня змінливості цін;

2) класифікація ризиків: "CryptoAnalyzer" може класифікувати різні рівні ризиків для кожної криптовалюти. Це допомагає інвесторам оцінити потенційні небезпеки та приймати обґрунтовані рішення щодо свого портфеля;

3) надсилання сигналів: бот може надсилати користувачам сигнали щодо змін волатильності або важливих подій на ринку. Це допомагає трейдерам залишатися в курсі актуальних подій;

4) оцінка тенденцій: "CryptoAnalyzer" може надавати користувачам інформацію про загальні тенденції на ринку криптовалют та рекомендації стосовно можливих стратегій;

5) персоналізовані рекомендації: на основі історії та поведінки користувача бот може надавати персоналізовані рекомендації щодо оптимального управління портфелем та ризиками.

Загальний успіх таких інтелектуальних систем полягає в їхній здатності швидко реагувати на зміни на фінансових ринках, надавати точний аналіз та забезпечувати зручний інтерфейс для користувачів. Однак, високоризиковані аспекти пов'язані з можливістю помилок у прогнозах та залежністю від надзвичайних ринкових умов. У цілому, ці успішні реалізації свідчать про потенціал інтелектуальних систем у фінансовому аналізі через Telegram-бот, використовуючи їхні переваги для вигоди та інформування користувачів.

Аналіз останніх досліджень вказують на ріст зацікавленості у вдосконаленні систем прогнозування та класифікації фінансового ринку, а також на розширення використання технологій чат-ботів, зокрема через платформу Telegram, для полегшення доступу та взаємодії з інтелектуальними системами.

Ці дослідження мають потенціал поліпшити якість прогнозів, роблячи їх більш точними та адаптивними до змін у ринкових умовах. При цьому важливо враховувати високий ризик, пов'язаний з непередбачуваністю фінансового ринку, і прагнути розвивати системи, які можуть ефективно пристосовуватися до динамічних змін, що й є задачею данної роботи.

1.2 Telegram: система миттєвого обміну повідомленнями

Системи миттєвого обміну повідомленнями (англ. Instant messaging, IM) – це тип телекомунікаційної послуги, яка дозволяє користувачам обмінюватися текстовими повідомленнями в режимі реального часу через Інтернет. IM-системи часто використовуються для особистого спілкування, але також можуть бути використані для професійних цілей, таких як групова робота або підтримка клієнтів.

Є багато різних систем миттєвого обміну повідомленнями, доступних на ринку. Деякі з найпопулярніших включають: WhatsApp, Facebook Messenger, Skype, Viber, Telegram, Slack та інші.

Однією з переваг систем миттєвого обміну повідомленнями є їхня простота використання. Користувачам потрібно лише завантажити програму та створити обліковий запис, щоб почати використовувати її. IM-системи також часто безкоштовні або мають низьку вартість.

Серед перелічених систем миттєвого обміну повідомленнями більш детально розглянемо Telegram. Він є одним із найпростіших у використанні IM-систем.

Історія Telegram почалася у серпні 2013 року, коли була випущена перша версія. Його розробка була ініційована прагненням створити безпечний та конфіденційний

інструмент для обміну повідомленнями. Протягом років платформа пройшла кілька оновлень та розширила свою користувацьку базу завдяки своїм особливостям та цінностям, які вона пропонує своїм користувачам. Перша версія додатку була запущена для iOS, а пізніше в тому ж році – для Android.

Telegram – це месенджер, який надає можливість обміну повідомленнями, відео- та аудіозаписами, фотографіями та іншими мультимедійними файлами. Заснований та розроблений підприємцем Павлом Дуровим та його братом Ніколаєм [1, 3].

Навіть не дивлячись на свій юний вік, Telegram наразі налічує понад 100 мільйонів активних користувачів щомісяця, особливо широко популярний в країнах Західної Європи. Розробники цього месенджера стверджують, що він володіє найкращим захистом серед аналогічних продуктів на ринку, але довіра користувачів грунтується не лише на цьому, а й на історії додатка та таланті його творців. На початку 2023 року в Telegram було понад 500 мільйонів активних користувачів у більш ніж 150 країнах.

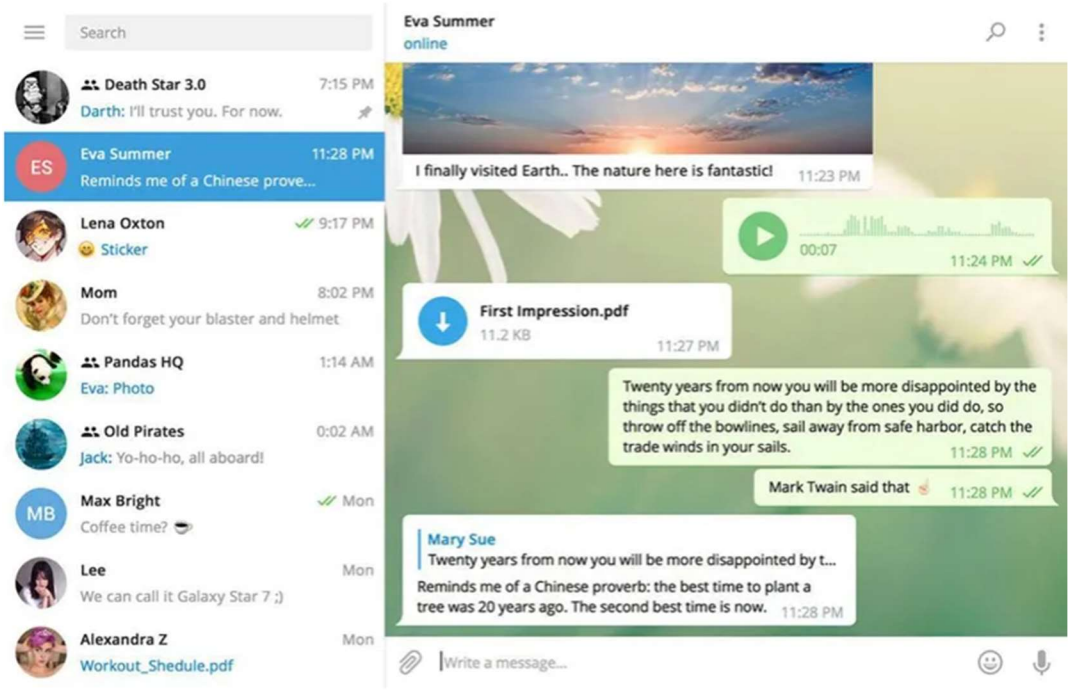


Рисунок 1.3 – Чати в Telegram

На сьогоднішній день, коли ми постійно оточені рекламою, все більш цінним стає спосіб спілкування, який не переривається рекламними повідомленнями. Таким способом спілкування є Telegram.

У Telegram немає реклами. Це означає, що користувачі можуть спілкуватися один з одним без перешкод від рекламних оголошень.

Telegram фінансується за рахунок продажу преміум-підписок. Преміум-підписка надає користувачам додаткові функції, такі як можливість завантажувати файли більшого розміру, використовувати більшу кількість стикерів і відеоповідомлень, а також мати доступ до більшої кількості каналів і груп.

Telegram відомий своєю фокусованістю на конфіденційності та безпеці користувачів. Він використовує шифрування для захисту пересиланих повідомлень та даних користувачів. Пріоритет безпеки в Telegrami визначається кількома ключовими аспектами, що описані нижче.

1. Шифрування E2E (end-to-end). Telegram використовує шифрування E2E для забезпечення безпеки основного змісту повідомлень. Це означає, що дані шифруються на пристрої користувача і розшифровуються лише на пристрої отримувача, що ускладнює можливість несанкціонованого доступу до персональної інформації.

2. Конфіденційні чати. Telegram дозволяє користувачам створювати конфіденційні чати, в яких повідомлення можуть бути самознищуватися через визначений час після читання. Це підвищує рівень приватності та захисту конфіденційної інформації.

3. Безпека віддалених сеансів. Telegram надає можливість перевірки та відключення активних сесій з інших пристроїв. Це дозволяє користувачам контролювати доступ до свого облікового запису та захищати його від несанкціонованого використання.

4. Точка доступу до облікового запису. Для забезпечення додаткового рівня безпеки, Telegram дозволяє встановлювати додатковий пароль для доступу до додатку або використання вбудованої системи розпізнавання обличчя/відбитка пальця.

5. Шифрування файлів. Усі медіа-файли, які відправляються через Telegram, також шифруються. Це включає в себе фотографії, відео, документи та інші типи файлів.

Ці заходи в сукупності створюють надійне середовище для обміну повідомленнями та контентом, забезпечуючи високий стандарт конфіденційності для користувачів Telegram. Крім цього, багатьом відомо, що у Telegrami є функція під назвою "Секретний чат", яка забезпечує максимальну конфіденційність. У цих чатах повідомлення шифруються від початку до кінця, а після закінчення заданого терміну видаляються. Спочатку розробники не додали цю функцію, оскільки вона була не дуже зручна: секретні чати залежать від пристрою, і розмову не можна продовжити на іншому пристрої. Проте пізніше розробники вирішили все ж таки додати цю функцію, оскільки вона стала більш популярною.

Особливістю Telegram також є можливість створення каналів, ботів, а також групового чатування [2].

Telegram-канали – це спосіб залучити нову аудиторію. Спочатку вони були створені користувачами для обговорення спільних інтересів, але потім їх почали використовувати ЗМІ для залучення дорослої аудиторії. Це все ще складне завдання, але Telegram пропонує кілька функцій, які можуть допомогти. Наприклад, Telegram дозволяє користувачам створювати групи до 200 000 учасників. Це ідеально підходить для спілкування з великою групою людей, наприклад, з членами спільноти або колегами [3].

У 2015 році Telegram створив окреме середовище для управління та створення ботів – Botfather. З тих пір було створено безліч ботів, у тому числі й універсальні чат-боти ЗМІ, які допомагають залучати мобільну аудиторію на їхні сайти. Навіть якщо

ви не є досвідченим програмістом, ви можете написати програму-помічника та використовувати її для побудови власної мережі. Для цього вам знадобляться лише базові знання використання інтерфейсів додатків.

Щоб створити бота, користувач повинен спочатку зв'язатися з BotFather. Для цього потрібно відкрити чат з BotFather і набрати команду /newbot. BotFather запропонує користувачеві ввести назву бота, його опис та псевдонім. Після того, як користувач надасть цю інформацію, BotFather створить бота та надасть користувачеві токен бота [1, 2].

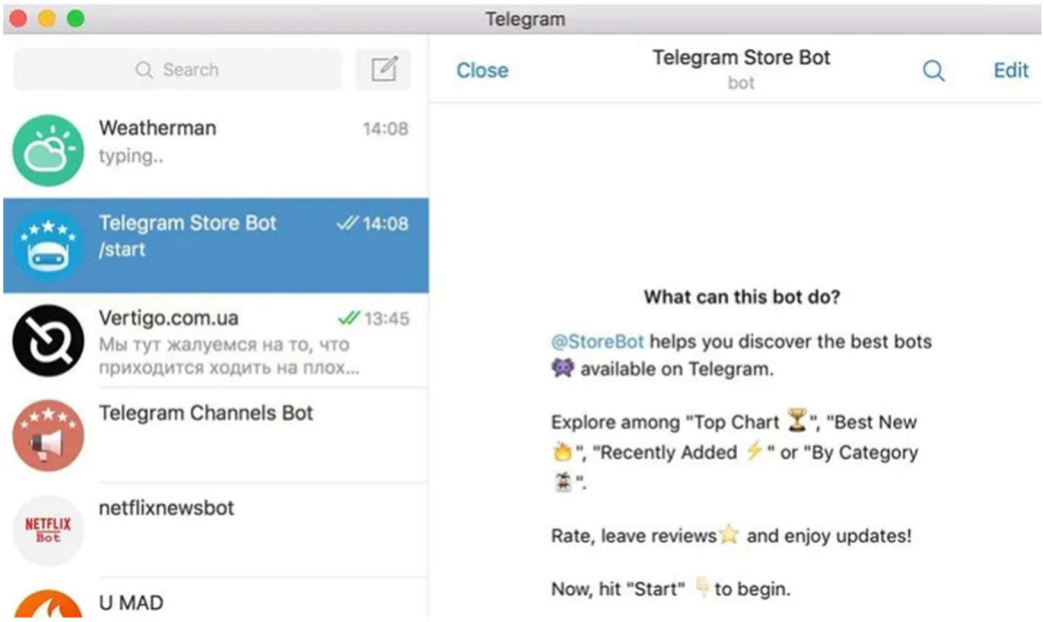


Рисунок 1.4 – Боти в Telegram

Токен бота – це унікальний ключ, який використовується для ідентифікації бота. Користувач повинен використовувати токен бота, щоб керувати ботом. Наприклад, щоб надіслати повідомлення боту, користувач повинен використовувати команду /sendmessage.

BotFather також дозволяє користувачам налаштовувати своїх ботів. Наприклад, користувачі можуть налаштувати:

- тип бота;
- відповіді бота;
- управління ботом.

Так як тема моєї магістерської роботи стосується одного з ключових показників фінансового ринку, то неможливо не додати інформацію стосовно того, що Telegram провів одну з найбільш вдалих кампаній Initial Coin Offering (ICO) для створення власної криптовалюти Gram у січні 2018 року. Завданням було залучити приблизно 1.7 мільярда доларів США через ICO.

ICO – це форма краудфандингу (вид збору коштів), яка використовує криптовалюту. Вкладники отримують токени в обмін на свої вкладення, які можна потім обміняти на продукти або послуги компанії.

Telegram Open Network (TON) був амбіційним блокчейн-проектом, який розроблявся командою Telegram для створення децентралізованої платформи та криптовалюти. Проте, через правові проблеми, цей проект був призупинений, залишивши значний слід в історії криптоспільноти та підкресливши важливість врахування юридичних аспектів при розробці та впровадженні блокчейн-проектів.

Як висновок, можна зазначити, що Telegram – це популярний месенджер, який пропонує ряд переваг перед іншими подібними продуктами. До них відносяться:

1) конфіденційність та безпека: Telegram визначається високим рівнем шифрування та можливістю автоматичного видалення повідомлень, що підвищує конфіденційність даних користувачів;

2) швидкість та ефективність: оптимізований для високої швидкості передачі повідомлень, Telegram забезпечує ефективну комунікацію без значних затримок;

3) відкритість і прозорість: застосування прозорого протоколу підкреслює відкритість та доступність системи для користувачів;

4) безкоштовність і відсутність реклами: відсутність вартості використання та відсутність інтегрованої реклами роблять Telegram доступним і бездоганно чистим від комерційних впливів;

5) без обмежень на обсяг повідомлень: відсутність чітких обмежень на розмір повідомлень і файлів, які можна надсилати, забезпечує більшу свободу в обміні інформацією;

6) глобальна система: розгортання серверів Telegram по всьому світу сприяє надійності та стабільності обслуговування.

Рисунок 1.5 – Telegram найзахищеніший безкоштовний месенджер

Ці переваги роблять Telegram привабливим вибором для користувача у сфері миттєвого обміну повідомленнями, підкреслюючи його унікальність і привабливість порівняно з іншими платформами.

1.3 Джерело інформації Yahoo Finance API

У процесі фінансового аналізу та досліджень інвестиційних можливостей важливою ланкою є доступ до актуальних та достовірних фінансових даних. У цьому контексті великою популярністю користується використання різноманітних API (інтерфейсів програмування застосунків), які забезпечують доступ до певного переліку інструментів та даних фінансового ринку.

Один із ключових учасників цього сегменту – Yahoo Finance [10]. Власне, Yahoo Finance володіє потужним та широким API, яке надає користувачам доступ до обширної бази фінансових даних, включаючи інформацію про волатильність, ціни акцій, графіки, новини та інші релевантні показники ринку.

В даному пункті необхідно розглянути та проаналізувати API від Yahoo Finance. Відслідкувати основні можливості цього інтерфейсу, його структуру та функціонал, а також оцінити, як цей інструмент може бути використаний для забезпечення точності та актуальності фінансового аналізу. Розглядання API від Yahoo Finance дозволить зрозуміти, як цей інтерфейс може вплинути на якість досліджень, а також як використовувати його для отримання необхідних даних та інформації в контексті вивчення та аналізу фінансових подій.

Тож, Yahoo Finance – це фінансовий портал, який надає різноманітні фінансові інструменти та послуги для інвесторів та трейдерів. Він є частиною екосистеми Yahoo, і його основною метою є надання користувачам доступу до актуальної інформації про фінансові ринки, новини та аналітики [9].

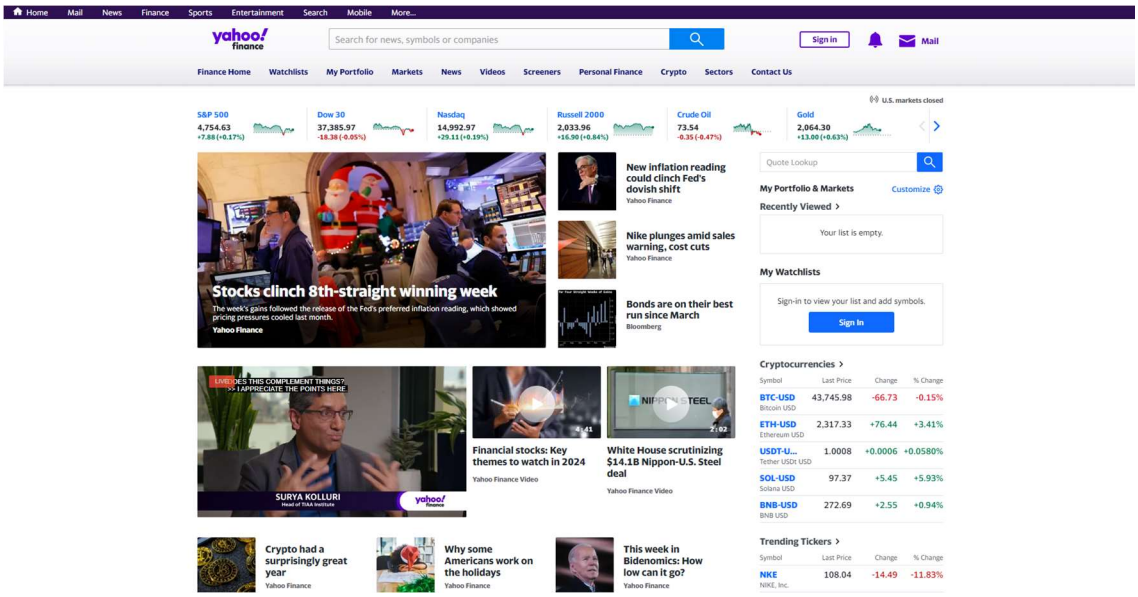


Рисунок 1.6 – API Yahoo Finance

За допомогою API розробники можуть використовувати фінансові дані Yahoo для створення власних додатків та аналітичних інструментів, зокрема портфелів акцій, цінових сповіщень та інших сервісів на основі даних.

API-клієнт надає доступ до широкого набору даних про фінансовий ринок. Дані доступні в різних форматах, включаючи JSON, XML і CSV.

Ось деякі з найпопулярніших методів Yahoo Finance API:

- getQuote – отримує поточну ціну акції;
- getHistoricalData – отримує історію цін акції;
- getTechnicalIndicators – отримує технічні показники для акції;
- getNews – отримує новини про акції.

Наприклад, для отримання поточної ціни акції Apple (AAPL) можна використовувати наступний код:

```
import yfinance as yf
# Отримати поточну ціну акції
quote = yf.Ticker("AAPL").info
# Вивести поточну ціну акції
```

```
print(quote["Last"])
     Для отримання історії цін акції "AAPL" за останній рік:
import yfinance as yf
# Отримати історію цін акції
data = yf.Ticker("AAPL").history(period="1y")
# Вивести історію цін акції
print(data)
```

В основі прогнозів на акції AAPL використовують дані історичної волатильності. Нижче – огляд історії цін акцій від Yahoo Finance (рис. 1.7).

Рисунок 1.7 – Огляд історії цін акцій AAPL

Yahoo Finance API пропонує широкий набір можливостей і функцій, включаючи історичні котирування, новини фондового ринку та вичерпні фінансових данних. Ці дані можна використовувати для створення власних фінансових додатків та аналітичних інструментів, таких як аналіз фондового портфеля, сповіщення про ціни на акції та інші сервіси, що базуються на даних.

Даний портал дозволяє розробникам отримувати доступ до фінансових даних з різних ринків і цінних паперів, включаючи акції, взаємні фонди, опціони та ф'ючерси. Маючи доступ до ринкових даних у режимі реального часу, розробники можуть

створювати інтерактивні графіки та торгові системи, здійснювати моніторинг портфелів та аналіз фондового ринку [9, 10].

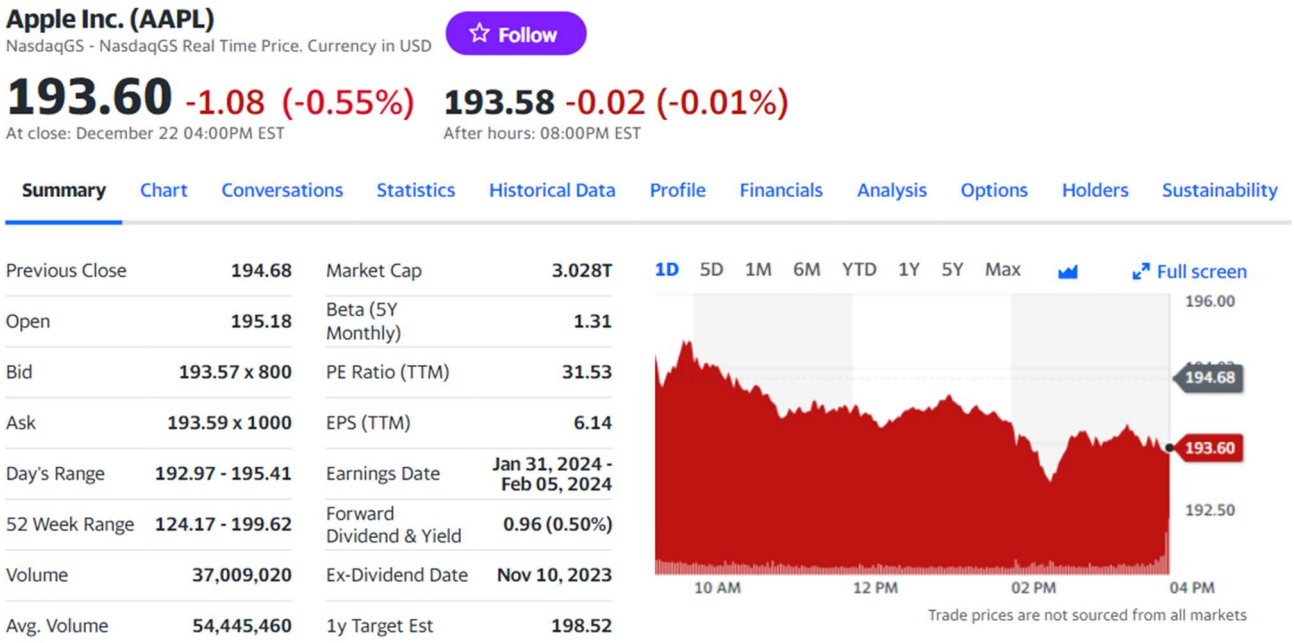


Рисунок 1.8 – Аналіз вартості акцій APPL у реальному часі

Деякі ключові компоненти та функції Yahoo Finance включають:

1) поточні ціни акцій і графіки: Yahoo Finance надає графіки з реальними чи затриманими цінами акцій, які дозволяють відстежувати рух цін і виявляти тенденції;

2) свіжі новини та статті про фінанси, компанії та економіку, які можуть вплинути на ринки;

3) фінансові дані та звіти: ви можете отримати доступ до фінансових звітів, балансів, звітів про прибуток та інших фінансових показників компаній;

4) портфель і відстеження: ви можете створити власний портфель акцій, слідкувати за його розвитком та отримувати сповіщення про зміни в цінах;

5) аналіз та інструменти: інструменти для технічного та фундаментального аналізу, такі як графіки, технічні індикатори та показники;

6) дані по індексам і ринках: інформація про головні фінансові індекси та глобальні ринки;

7) інформація та дані щодо опціонів та ф'ючерсів.

Yahoo Finance API відіграє ключову роль у створенні телеграм-бота, надаючи різнобічні можливості та функції. Однією з переваг є отримання фінансових даних, включаючи поточні та історичні дані про ціни акцій, обсяги торгів і фінансові звіти компаній.

API дозволяє створювати графіки та візуалізації для аналізу цінових тенденцій та ринкової динаміки. Це важливий інструмент для інвесторів, які шукають глибше розуміння ринкових тенденцій. Крім того, API може служити основою для створення засобів моніторингу та сповіщень, що сприяє вчасному виявленню та реагуванню на зміни в портфелі акцій.

Інтеграція з API дозволяє отримати доступ до різноманітних фінансових показників, що полегшує глибший фундаментальний аналіз компаній та ринків. Бот може використовувати цю інформацію для реалізації функцій інвестиційного порадника, допомагаючи користувачам приймати обгрунтовані рішення щодо їхніх інвестицій.

Крім того, інтеграція з API сприяє створенню інтерактивних інтерфейсів, що полегшують взаємодію користувачів з фінансовою інформацією. Такі інтерфейси можуть надавати розширені можливості запитань та відповідей, забезпечуючи зручний інструмент для аналізу та спостереження за фінансовим ринком.

Для більшої деталізації, хочу навести кілька конкретних прикладів того, як Yahoo Finance API може використовуватися для створення телеграм-бота.

Першим прикладом виступає випадок, коли бот може використовувати Yahoo Finance API для надання користувачам оновлень про поточну ціну акцій у реальному часі, створюючи можливість оперативно відстежувати зміни на ринку та приймати швидкі рішення.

Крім того, використовуючи API для запиту даних про історію цін, технічні показники та іншу інформацію, бот може забезпечити користувачів аналітичними звітами, дозволяючи їм глибше розуміти ринкові тенденції та приймати обгрунтовані інвестиційні рішення.

Застосовуючи API для запиту даних про новини та події, бот може також повідомляти користувачів про важливі події на фінансовому ринку, що допомагає тримати їх у курсі останніх змін та впливати на їхні інвестиційні стратегії. Такий підхід створює повний та інформативний інструмент для аналізу та моніторингу фінансового ринку прямо у чат-платформі.

Звичайно, Yahoo Finance API не є єдиним API, який можна використовувати для створення телеграм-бота. Існує також ряд інших API, які можуть забезпечити доступ до схожих даних. Однак Yahoo Finance API є одним з найпопулярніших і найпростіших у використанні API, а що саме головне – безкоштовним.

Для реалізації телеграм-бота фінансового ринку можна використовувати різноманітні фінансові показники та інформацію, яка надається фінансовими ресурсами. Ці показники та інформація можуть використовуватися для надання користувачам різноманітних аналітичних звітів, сповіщень та інших послуг.

Ось деякі ключові показники, які можна включити в такий бот:

1) ціна акції (Stock Price): поточна ціна акції обраної компанії або фінансового інструменту;

2) графік цінових змін (Price Chart): графічне відображення динаміки цін для кращого розуміння тенденцій;

3) доходність дивідендів (Dividend Yield): відношення річних дивідендів на акцію до ціни акції;

4) фінансові новини (Financial News): актуальні новини та статті, пов'язані зі світовими фінансами;

5) волатильність (Volatility): міра ступеня коливань цін акцій або фінансового інструменту за певний період часу;

6) стабільність цін (Beta): міра того, як ціни акцій реагують на рухи ринку;

7) показник цінності (PE Ratio): відношення ціни акції до прибутку на акцію;

8) індекси ринку (Market Indices): значення ключових фінансових індексів, таких як S&P 500 чи Dow Jones;

Проте потрібно розуміти, що вибір конкретних фінансових показників та інформації, які будуть використовуватися для реалізації телеграм-бота, залежить від цілей бота. Наприклад, якщо бот призначений для надання користувачам оновлень про поточні ціни акцій, то він буде використовувати такі показники, як ціна та об'єм торгівлі. Якщо бот призначений для надання користувачам аналітичних звітів про певні акції, то він буде використовувати такі показники, як історія цін, технічні показники та новини.

Телеграм-бот фінансового ринку може бути потужним інструментом для користувачів, які хочуть отримати доступ до актуальної фінансової інформації та аналітики. Вибір правильних фінансових показників та інформації є ключовим для створення бота, який буде корисним і інформативним для користувачів.

В рамках моєї реалізації телеграм-бота основний акцент буде зроблено саме на волатильності валют. Цей ключовий показник відображатиме ступінь коливань цін акцій та фінансових інструментів протягом конкретного періоду часу. Волатильність стане важливим фактором для інвесторів та трейдерів, надаючи їм інформацію про рівень ризику та потенційні можливості отримання прибутку.

Графічне представлення динаміки волатильності та аналіз даного показника стане важливою частиною функціоналу бота, сприяючи більш інформованим та обґрунтованим фінансовим рішенням користувачів в умовах змінливого ринкового середовища.

Yahoo Finance API надає дані для різних показників волатильності. Найпоширенішим показником є історична волатильність (Historical Volatility), яка представляє стандартне відхилення цін за певний період (наприклад, 1 місяць, 3 місяці, 1 рік). Також доступні інші показники, такі як Beta (міра залежності активу від загального ринку) та ATR (Average True Range - середній справжній діапазон).

Ось як можна додати волатильність валют до ключових показників для фінансового бота:

```
import requests
import json
# Визначте валюти, які ви хочете включати в бот
currencies = ["EURUSD", "USDJPY", "GBPUSD"]
# Виберіть показник волатильності, який ви хочете використовувати
indicator = "hv"
# Використовуйте Yahoo Finance API для отримання даних про волатильність валют
for currency in currencies:
    response = requests.get(
        "https://finance.yahoo.com/quote/{}/{}".format(currency, indicator)
    )
    data = json.loads(response.content)
# Додайте дані про волатильність валют до вашого бота
print("{}: {}".format(currency, data["quote"]["financialData"]["pastMonthVola
tility"]))
```

Отже, волатильність є важливим показником, який може бути використаний для підвищення ефективності фінансового бота. Додавши волатильність до ключових показників, ви зможете надати своїм користувачам більш повну картину фінансового ринку та допомогти їм прийняти більш обґрунтовані інвестиційні рішення.

1.4 Волатильність як ключовий показник фінансового ринку

1.4.1 Показник волатильності та його застосування

Волатильність – це показник, який характеризує мінливість ціни активу протягом певного періоду часу [17, 19]. У фінансовій сфері волатильність найчастіше використовується для оцінки ризиків, пов'язаних з інвестиціями в різні фінансові інструменти, зокрема валюти.

Зростання волатильності може вказувати на підвищення ризику та нестабільність, тоді як зменшення може свідчити про більш стабільні умови на ринку.

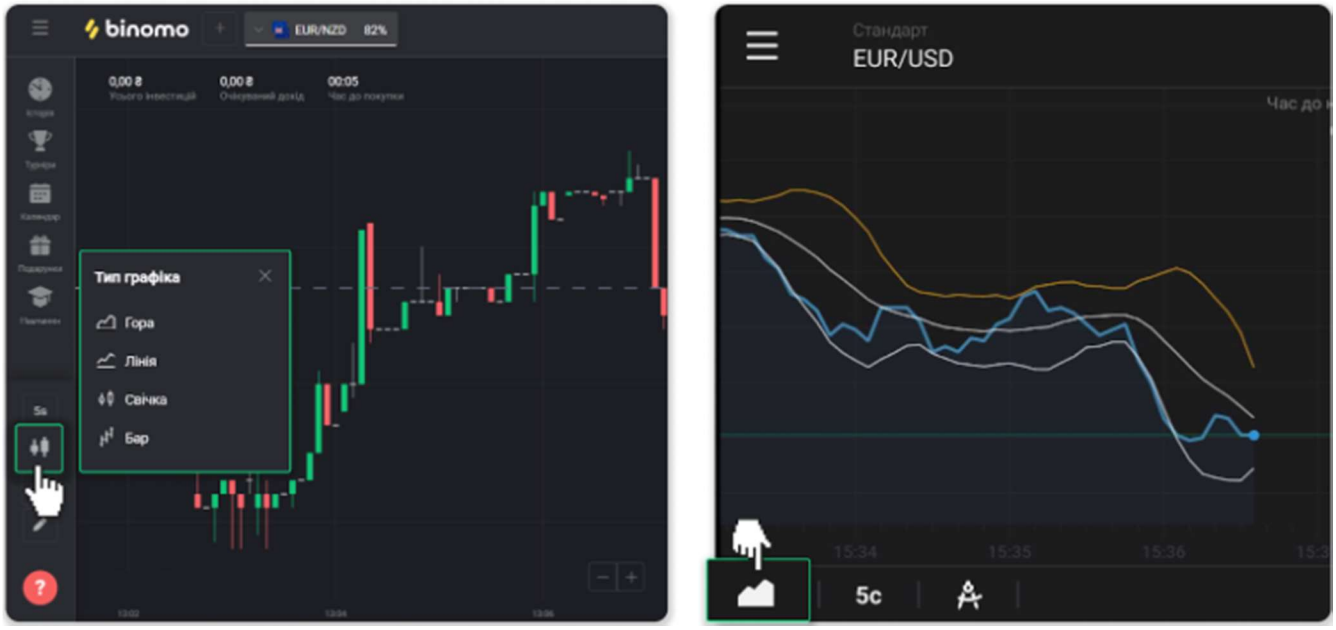


Рисунок 1. 9 – Графік зміни ціни активу

Зростання волатильності часто спостерігається під час важливих подій, таких як економічні кризи чи політичні нестабільності.

Наприклад, волатильність валют була високою протягом усього 2023 року. Це було викликано низкою факторів, включаючи війну в Україні, інфляцію та зростання процентних ставок.

Волатильність може бути виміряна різними способами, такими як стандартне відхилення курсу валюти від середнього значення протягом заданого періоду часу, історична волатильність чи імпліцитна волатильність, яка визначається опціонами на цінні папери.

Формула для розрахунку стандартного відхилення виглядає так:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{1.1}$$

де σ – стандартне відхилення;

$\sum$ – сума;

x – значення точки даних;

μ – середнє значення;

N – кількість точок даних.

Як приклад, інвестор може використовувати формулу для розрахунку стандартного відхилення, щоб визначити, чи потрібно йому диверсифікувати свій портфель.

Якщо стандартне відхилення портфеля інвестора високе, то це означає, що його портфель схильний до ризику. Інвестор може диверсифікувати свій портфель, додавши до нього активи з низьким стандартним відхиленням. Це допоможе йому знизити ризик свого портфеля.

Існують кілька типів волатильності валют [18], які використовуються для вимірювання рівня коливань курсів валют на фінансових ринках. Основні типи волатильності описані нижче.

1. Історична волатильність (Historical Volatility) – вимірює коливання цін у минулому на основі статистичного аналізу цінової динаміки. Зазвичай виражається у відсотках і використовується для оцінки ризику.

2. Реалізована волатильність (Realized Volatility) – є конкретною мірою коливань цін за певний період часу. Вона базується на фактичних цінах активу та використовується для оцінки волатильності в певний момент.

3. Прогнозована волатильність (Implied Volatility) – визначається на основі цін опціонів на фінансовому ринку. Прогнозована волатильність вказує на очікувані коливання цін у майбутньому, як визначається учасниками ринку.

4. Статистична волатильність (Statistical Volatility) – вимірюється за допомогою статистичних методів аналізу цін, таких як середньоквадратичне відхилення. Цей метод базується на статистичних показниках та аналізі цінових патернів.

5. Абсолютна волатильність (Absolute Volatility) – це вимірювання коливань цін в абсолютних величинах, а не відсотках. Вона дозволяє оцінити конкретні величини змін цін в одиницях валюти.

Волатильність валют є ключовим показником. Цей показник актуальний для різних учасників ринку, включаючи інвесторів та підприємства, які здійснюють транснаціональну торгівлю. Інвестори використовують інформацію про волатильність для управління ризиками та прийняття обґрунтованих рішень щодо свого портфеля. Підприємства планують свої бізнес-стратегії, враховуючи можливі впливи коливань валютних курсів. Глобальні події та нестабільність в світі можуть збільшити волатильність. Трейдери використовують цей показник для розробки торгових стратегій. Центральні банки реагують на волатильність при формуванні грошової політики. Загалом, волатильність валют визначає фінансовий ландшафт, важливий для рішень учасників ринку в умовах постійних змін та невизначеності.

Отже, виходячи з цього всього, можна прийти до висновку, що дана тема відкриває можливості для розширення знань у галузі фінансового аналізу та розвитку інноваційних інструментів для мінімізації ризиків на фінансових ринках. Результати подальших досліджень в цьому напрямку можуть стати важливим внеском в

покращення стратегічного прийняття рішень та функціональності фінансових інструментів у сучасному інвестиційному середовищі.

Для розуміння основних факторів, що впливають на волатильність, важливо розглянути різноманітні аспекти, такі як економічні, політичні та інші фундаментальні чинники. Оптимальне вивчення цих факторів дозволяє інвесторам приймати обдумані рішення та ефективно управляти своїм портфелем в умовах змінливого ринкового середовища. Розуміючи причини, що спричиняють різке рухання будь-якої валютної пари, можна вчасно скористатися цим для отримання прибутку.

Волатильність валютних пар на фінансових ринках піддається впливу різноманітних факторів. Ось деякі з них:

1) економічні новини та події: публікація важливих економічних показників, таких як ВВП, зайнятість, інфляція, може призвести до значущих змін у вартості валютних пар. Давайте розглянемо приклад. 15 березня 2023 року, незважаючи на негативні дані щодо індексу цін виробників у США та виробничого індексу Empire Manufacturing USD [20], курс євро до долара США несподівано зміцнився. Явних передумов для такого руху не існувало – економічні показники вказували на ослаблення долара. Однак у цей раз вплив виник від банку Credit Suisse, чиї акції 15 березня різко впали більш ніж на 25% через потенційне банкрутство. Цей несподіваний обвал акцій Credit Suisse став каталізатором для зниження цін на долар і подальшого зміцнення євро;

Рисунок 1.10 – Зміцнення курсу євро до долара США

2) геополітичні події: конфлікти, вибори, санкції та інші глобальні події можуть викликати нестабільність на фінансових ринках, впливаючи на волатильність валют;

Рисунок 1.11 – Вплив геополітичних подій на волатильніст валют

3) центральні банки та грошова політика: рішення центральних банків щодо процентних ставок, кредитування та інших монетарних інструментів можуть суттєво впливати на валютні пари;

4) технічний аналіз: зміни в цінах та обсягах торгів, а також технічні сигнали можуть стимулювати торгові рішення та впливати на волатильність;

5) процентні ставки: різниця у процентних ставках між різними країнами може впливати на привабливість валют для інвесторів та торгівлю;

6) попит і пропозиція: зміни в обсягах торгів та сприйняттям валют можуть виникнути через різні фактори, такі як великі торгові угоди або зміни в економічному середовищі;

7) спекуляції та трейдерське становище: рішення трейдерів, їхні очікування та стратегії також можуть сприяти волатильності на ринку;

8) погода та природні катастрофи: деякі країни, зокрема ті, що спеціалізуються на виробництві сільськогосподарської продукції, можуть відчути вплив природних катастроф на вартість їхньої валюти.

Наприклад, в серпні 2019 року лісові пожежі в Австралії суттєво ударили по експорту країни. По закінченню року, внаслідок виходу статистики та загострення економічних проблем, а також врахування початку пандемії, в березні 2020 року валютна пара AUD/USD досягла мінімуму за останні 17 років [20]. Цей приклад ілюструє, як непередбачувані події можуть суттєво вплинути на валютний ринок та вартість валют.

Рисунок 1.12 – Вплив пандемії на волатильніст валютної пари

9) торговельні угоди та міжнародні відносини: укладання або скасування торгових угод між країнами може створювати нестабільність у валютному ринку.

Ці фактори взаємодіють, і їхні впливи можуть бути складними та непередбачуваними.

1.4.2 Аналіз найбільш поширених валютних пар за волатильністю

Волатильність є важливим аспектом для трейдерів та інвесторів, оскільки вона вказує на ризик і можливості отримання прибутку на валютному ринку. Валютні пари з високою волатильністю можуть демонструвати швидкі та значні зміни цін, що може бути як перевагою, так і ризиком для учасників ринку, залежно від їхніх стратегій та цілей.

Волатильність валютних пар є ключовим аспектом фінансового ринку, оскільки вона визначає рівень коливання цін. Цей показник вказує на амплітуду рухів цін за певний період часу. Наприклад, за 4 години ціна однієї пари може змінитися на 30 пунктів, а ціна іншої пари весь цей час може перебувати в межах 10 пунктів.

Це означає, що на першій парі трейдери мають потенціал заробити більше грошей, якщо правильно вгадають напрямок руху ціни. Однак вони також можуть швидше втратити гроші, якщо помиляться.

Тобто, якщо трейдер відкриває позицію на повному стандартному лоті по парі EUR/USD, при ціні пункту 10 USD, то він може заробити 300 USD, але за умови, що ціна зросте в його бік на 30 пунктів. Однак він також може зазнати збитку.

Тому трейдерам важливо оцінювати ризики та потенційні прибутки перед відкриттям позиції на валютній парі.

Деякі валютні пари можуть бути більш волатильними, що вигідно використовувати для активної торгівлі, а інші – менш волатильними, що може бути корисним для тих, хто віддає перевагу стабільності. Управління ризиками та врахування волатильності допомагають ефективно керувати портфелем та уникнути непередбачуваних збитків.

Розглянемо наглядний приклад мінливості ціни на фіксованому часовому інтервалі.

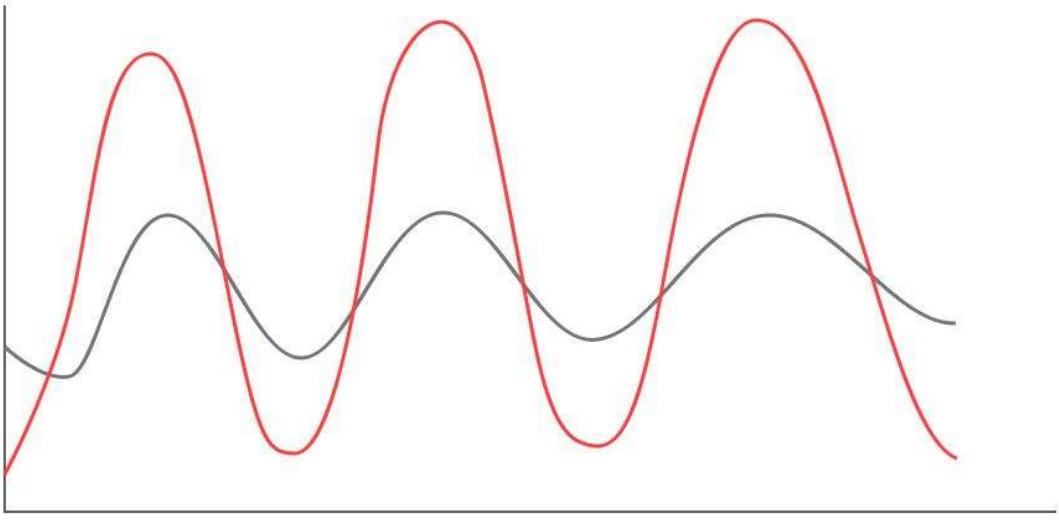


Рисунок 1.13 – Графічне відображення мінливості ціни на фіксованому часовому інтервалі

На порівняно однаковому часовому проміжку амплітуда руху у червоної цінової лінії більше – цей актив волатильніший. На ньому можна більше заробити, але в той же час є пов'язані з високою волатильністю ризики – торкання стопів, прослизання.

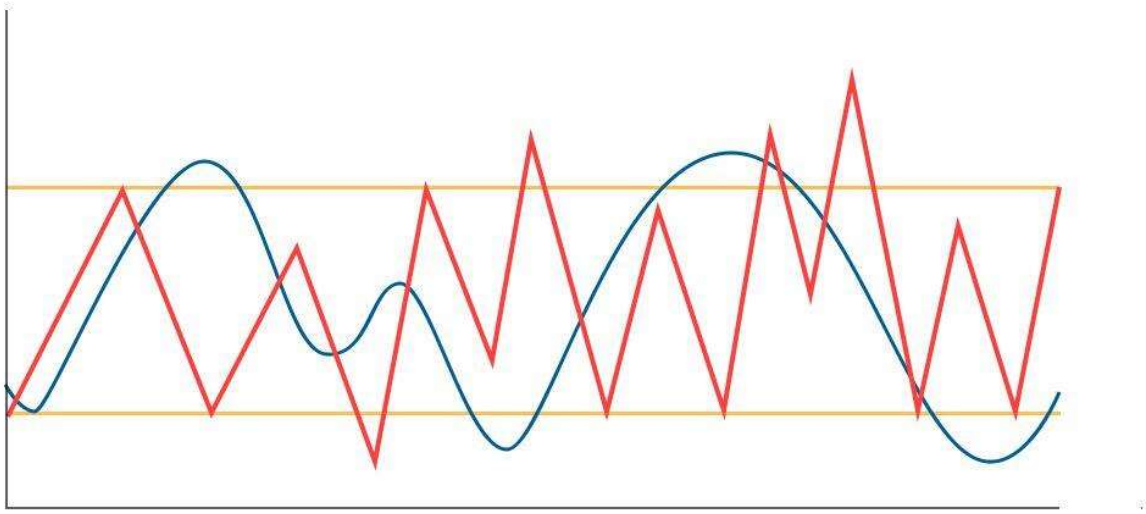


Рисунок 1.14 – Графічне прослідковування волатильності активу

В одному коридорі, частота торкання червоної лінії є вищою, що свідчить про те, що вона частіше перетинає й виходить за межі коридору. Це вказує на більшу волатильність цього активу.

На свічковому графіку волатильність можна визначити за розміром свічок та зміною цін у відсотковому вираженні. Якщо з'являється свічка з великим тілом і довгими тінями порівняно з попередніми свічками на графіку, це свідчить про зростання волатильності.

Важливо розглядати волатильність в парі з ліквідністю. Ліквідність вказує на те, наскільки легко можна купити або продати актив (наприклад, валюту, цінні папери чи інші фінансові інструменти) без значних втрат вартості. Висока ліквідність означає, що актив може бути швидко куплений або проданий на ринку без значних змін у його ціні.

Найбільш волатильні пари на ринку Forex часто є екзотичними, але через їхню низьку ліквідність торгувати ними може бути складно, особливо для початківців. З іншого боку, пара EUR/USD є найбільш ліквідною, має високу волатильність і користується значною популярністю серед трейдерів. Тож розглянемо пари з урахуванням їхньої ліквідності.

Основні волатильні валютні пари – це так звані «мажори», найбільш ліквідні і мінливі пари, що часто використовуються. До них відносяться валюти розвинених країн із сильною економікою: USD, EUR, GBP, CHF, CAD, AUD та JPY. Відповідно, основні валютні пари – це пари з USD.

		Пипсы	%
○	AUD/CAD	83.11	0.91
○	AUD/CHF	64.04	1.02
○	AUD/JPY	115.78	1.27
○	AUD/NZD	86.27	0.82
○	AUD/USD	84.08	1.25
○	CAD/CHF	56.02	0.82
○	CAD/JPY	117.42	1.18
○	CHF/JPY	157.59	1.08
○	EUR/AUD	149.69	0.95
○	EUR/CAD	104.09	0.72
○	EUR/CHF	62.78	0.64
○	EUR/GBP	57.44	0.66
○	EUR/JPY	161.18	1.12
○	EUR/NZD	166.16	1.00
◉	EUR/USD	88.56	0.84
○	GBP/AUD	161.45	0.89
○	GBP/CAD	128.57	0.77
○	GBP/CHF	85.39	0.75
○	GBP/JPY	182.25	1.10
○	GBP/NZD	176.91	0.93

		Пипсы	%
○	GBP/USD	114.91	0.95
○	NZD/JPY	106.18	1.22
○	NZD/USD	78.74	1.23
○	USD/BRL	729.56	1.37
○	USD/CAD	95.10	0.70
○	USD/CHF	81.35	0.87
○	USD/CNY	265.71	0.38
○	USD/DKK	566.79	0.81
○	USD/HKD	65.68	0.08
○	USD/ILS	434.48	1.25
○	USD/INR	342.56	0.41
○	USD/JPY	158.74	1.17
○	USD/MXN	2,169.79	1.10
○	USD/RUB	11,323.60	1.75
○	USD/SEK	1,317.79	1.27
○	USD/SGD	74.59	0.55
○	USD/TRY	0.00	0.00
○	USD/ZAR	2,617.95	1.48
○	XAG/USD	645.25	2.78
○	XAU/USD	2,678.50	1.49

Рисунок 1.15 – Основні валютні пари на Forex

За даними калькулятора, найбільш волатильними валютними парами протягом 20 тижнів є:

1) AUD/USD (Австралійський долар/Долар США): Австралія, яка практично не втручається в геополітичні конфлікти, має валюту, що сильно залежить від експорту, особливо враховуючи його вразливість до природних катаклізмів. Це може призводити до глибоких просадок її курсу в порівнянні з доларом США. Найкращим часом для торгівлі цією валютною парою на ринку Forex є Лондонська та Американська сесії;

2) USD/JPY (Долар США/Японська єна): про цю валютну пару кажуть, що її прогнозування за допомогою технічного аналізу є викликом через жорстке напівручне управління курсом з боку Центрального банку Японії. Основні торгові стратегії з USD/JPY включають внутрішньоденні операції та торгівлю на прорив рівнів на початку азіатської сесії. Також, через особливості процентних ставок, ця пара часто використовується в стратегії керрі-трейд;

3) GBP/USD (Британський фунт стерлінгів/Долар США): ця валютна пара є третьою за популярністю на ринку Forex, займаючи близько 12% всіх обсягів торгів. Висока волатильність пов'язана з монетарною політикою Федеральної резервної системи (ФРС), яка прагне стримати зростаючу інфляцію, а також з особливостями монетарної політики банку Англії та нюансами Brexit. Велика Британія, відома своїми фінансовими центрами, де більшість трейдерів працюють з національною валютою, має значні торгові обсяги, що призводить до великої мінливості цін. Найкращий час для торгівлі – Європейська торгова сесія.

Екзотичні валютні пари – це комбінації валют країни з розвиваючоюся економікою та країни з розвиненою економікою. Такі пари майже відсутні на міжнародних валютних ринках, оскільки економіки цих країн часто закриті і управляються вручну. Прогнозування та класифікація курсів валют екзотичних пар

ускладнюється відсутністю загальнодоступної фундаментальної інформації, навіть якщо йдеться про валюти ринків, що розвиваються.

Найбільш нестабільні екзотичні валютні пари на даний момент включають:

1) USD/RUB (Долар США/Російський рубль): російський рубль став лідером за нестабільністю через геополітичні турбуленції та санкції, які вплинули на бюджет росії. Втім, Центральний банк росії втрутився у ситуацію, використовуючи друкарський верстат та контролюючи курс;

2) USD/BRL (Долар США/Бразильський реал): політична напруженість та непередбачуваність монетарної політики призвели до постійних коливань цієї валютної пари;

3) USD/SEK (Долар США/Шведська крона): шведська крона, на відміну від швейцарського франка, є менш популярною валютою, не є вільно конвертованою та не використовується як захисний актив.

Розглядаючи дане питання більш глибше, було встановлено, що UAH та USD також є екзотичними валютними парами, оскільки Україна та росія не є членами групи 20 найбільших економік світу. Ці валютні пари (USD/RUB, USD/UAH) є нестабільним через такі фактори:

- економічні фактори: Україна та росія є країнами з нестабільним економіками з високим рівнем інфляції, безробіття та політичної нестабільності. Це фактори, які можуть призвести до коливань валютних курсів;
- політичні фактори: Україна та росія є країнами, які переживають політичну кризу. Це може призвести до коливань валютних курсів, оскільки трейдери намагаються оцінити вплив політичних подій на економіку країни;
- глобальні фактори: глобальні фактори, такі як зміни процентних ставок у Сполучених Штатах або іншій великій економіці, також можуть впливати на валютні курси екзотичних валют.

Гривня різко впала в 2022 році через російське вторгнення в Україну. Долар США став більш привабливим активом для інвесторів, оскільки він вважається більш безпечним.

Трейдери, які торгують валютними парами USD/UAH та USD/RUB, повинні бути готові до високих ризиків. Ці пари можуть принести великі прибутки, але також можуть призвести до збитків величезного масштабу.

Розглядаючи питання найменш волатильних валютних пар, можна встановити, що низька волатильність на ринку Forex пов'язана з економічною стабільністю країн і їхньою стійкістю до різноманітних збурень. Валюти з низькою волатильністю можуть слугувати захисними активами. Інший варіант - одна з валют постійно дешевшає в порівнянні з іншою.

Приклади найменш волатильних валютних пар:

1) USD/INR (Долар США/Індійська рупія): індійська економіка перебуває в глибокій кризі, поглибленій пандемією та зростанням цін на нафту. Індія є одним із найбільших імпортерів нафти, і це додає тиск на економіку. Інвестори виходять з країни, що призводить до стабільного зниження вартості індійської рупії в парі з доларом США. У цій парі волатильність невисока, вказуючи на стабільний висхідний тренд, де рупія здешевлюється протягом тривалого часу;

2) EUR/CHF (Євро/Швейцарський франк): швейцарський франк вважається символом стабільності, незважаючи на проблеми банківської системи. У парі з євро волатильність менша, ніж у парі з доларом США, завдяки географічному сусідству та тісним економічним зв'язкам Швейцарії і ЄС;

3) EUR/GBP (Євро/Фунт стерлінгів): незважаючи на вихід Великої Британії з ЄС, між обома країнами залишаються тісні економічні зв'язки. Це сприяє стабільності обох валют, і волатильність у цій парі залишається низькою, що вигідно впливає на обидві економіки.

Висновки до розділу 1

У першому розділі були ретельно розглянуті ключові аспекти, які становлять технологічні та методологічні тенденції у розробці інтелектуальних систем. Зокрема, досліджено використання технології чат-ботів на платформі Telegram як ефективного засобу миттєвого обміну повідомленнями. Приділена увага джерелу інформації Yahoo Finance API, що виступає важливим джерелом фінансових даних для подальшого використання у розробці інтелектуальної системи.

Особливий акцент був зроблений на важливому фінансовому показнику - волатильності валют. Розглянута та обгрунтована актуальність використання цього показника у контексті магістерської роботи, що спрямована на створення інтелектуальної системи класифікації волатильності валют. Визначено, що волатильність є ключовим елементом для інвесторів та трейдерів, надаючи інформацію про ризики та потенційні можливості на фінансових ринках.

Постановка задачі визначила конкретні кроки та завдання, які необхідно вирішити для успішної реалізації інтелектуальної системи, зокрема збір та обробку даних, визначення ознак впливу на волатильність, розробку моделей машинного навчання, інтеграцію з Telegram, а також оптимізацію та надання рекомендацій користувачам.

Таким чином, перший розділ визначив основний показник волатильності валют як ключовий елемент дослідження та постановки завдань у контексті розробки інтелектуальної системи.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ЗАДАЧІ КЛАСИФІКАЦІЇ

У другому розділі ми глибше розглянемо питання класифікації та методів машинного навчання, спрямованих на вирішення задачі класифікації волатильності валютних пар. Починаючи з загальної теми класифікації та розглядаючи ключові методи, такі як наївний баєсівський класифікатор (NB) та метод опорних векторів (SVM), ми також вивчимо використання показника волатильності та його значення у контексті аналізу фінансових ринків. Детально розглянемо застосування машинного навчання для класифікації рівнів волатильності валют.

2.1 Класифікація

Класифікація – це процес призначення або розподілу об'єктів до певних груп або категорій відповідно до їхніх характеристик чи ознак [21].

Цей метод використовується для встановлення структури або порядку в наборі даних, спрощуючи їх розуміння та дозволяючи ефективніше взаємодіяти з інформацією.

Класифікація застосовується в різних галузях, включаючи машинне навчання, біологію, економіку і багато інших сфер, де важливо впорядковувати чи групувати об'єкти з подальшим аналізом. Вона служить ефективним інструментом для систематизації об'єктів або явищ на основі їхніх спільних ознак, що сприяє кращому розумінню та впорядкуванню інформації.

Наприклад, у сфері машинного навчання, класифікація використовується для навчання моделей розпізнавання образів чи прогнозування, а в біології – для класифікації видів та вивчення їхніх взаємозв'язків. В економіці, класифікація може допомагати ранжувати компанії за рівнем прибутковості, а в інтернет-магазинах – для категоризації товарів та полегшення навігації. В освіті, цей підхід допомагає

студентам організовувати знання та розуміти основні концепції. Важливість класифікації полягає в її здатності до систематизації різноманітності об'єктів чи явищ, що сприяє ефективнішому аналізу та прийняттю рішень у різних сферах життя.

В сучасному світі роль класифікації стає все більше визначальною, особливо у галузі машинного навчання. Машинне навчання використовує алгоритми класифікації для розв'язання різноманітних завдань, починаючи від фільтрації спаму та закінчуючи розпізнаванням об'єктів на фотографіях.

У машинному навчанні завдання класифікації – це задача навчання з вчителем, в якому модель навчається на наборі даних.

Щодо типів класифікацій, то їх можна поділити на два основних типи [21, 22]:

1) за типом вхідних даних:

- описи ознак. Класифікація на основі текстових даних, таких як електронні листи, статті, код тощо;
- часові ряди. Класифікація на основі даних, які змінюються з часом, таких як ціни на акції, погодні дані тощо;
- матриця відстаней. Класифікація на основі даних, які можна представити у вигляді матриці відстаней, таких як зображення тощо;

2) за типом класів:

- нечіткі класи. Класи, які не мають чітких меж, таких як класи "добро" і "зло";
- класи, які перетинаються. Класи, які мають спільні елементи, такі як класи "хлопчик" і "дівчинка";
- двокласова класифікація. Класифікація на два класи;
- багатокласова класифікація. Класифікація на більше ніж два класи.

Розглянема декілька прикладів відомих компаній, які впроваджують класифікацію у своїх проектах:

1) компанія Google використовує класифікацію для розпізнавання об'єктів на своїх фотографіях. Це дозволяє Google ідентифікувати об'єкти на фотографіях, що робить їх корисними для пошуку, навігації та інших додатків;

2) компанія Amazon використовує класифікацію для категоризації продуктів на своєму вебсайті. Це дозволяє клієнтам легко знаходити продукти, які вони шукають;

3) компанія Netflix використовує класифікацію для рекомендацій фільмів та телешоу своїм користувачам. Це дозволяє Netflix пропонувати користувачам контент, який їм сподобається.

Класифікація та прогнозування – це два важливі завдання машинного навчання. Вони обидва використовуються для аналізу даних та прийняття рішень, але вони мають різні цілі та методи.

Таблиця 2.1 – Таблиця відмінностей між класифікацією та прогнозуванням

Характеристика	**Класифікація**	**Прогнозування**
Ціль	Розподіл об'єктів на групи	Передбачення майбутнього значення змінної
Дані	Дані про минулі та поточні значення змінної	Дані про минулі значення змінної
Вихід	Класифікація об'єкта	Значення змінної
Приклади	Розпізнавання образів, розпізнавання мови, класифікація клієнтів	Прогнозування цін на акції, прогнозування погоди, прогнозування ризику кредитування

Класифікація зазвичай є більш чіткою задачею, ніж прогнозування. Наприклад, класифікація фінансових інструментів може бути подібною до класифікації тварин. Припустимо, ми хочемо класифікувати акції компаній на фінансовому ринку.

Зазвичай, це можна зробити, визначивши їхню приналежність до різних секторів економіки, таких як технології, фінанси, охорона здоров'я тощо.

Однак, прогнозування конкретного показника, наприклад, зростання ціни акцій через рік, може бути більш складною задачею, аналогічно до прогнозування витрат на утримання собаки. Для цього може знадобитися аналіз різноманітних факторів, таких як фінансові звіти компанії, тенденції на ринку, макроекономічні показники та інші.

Таким чином, класифікація може визначити, до якого сектору відноситься компанія (аналогічно "собака" або "кішка"), але прогнозування конкретних фінансових показників може вимагати більш складного аналізу та використання прогнозуючих моделей у машинному навчанні.

Обрання класифікації для теми "Інтелектуальна система класифікації волатильності валют" є повністю обґрунтованою, тому що волатильності валют можна класифікувати на різні типи. Наприклад, представлений показник можна класифікувати за такими ознаками:

- за періодом часу волатильності можуть бути короткостроковими, середньостроковими або довгостроковими;
- за інтенсивністю волатильності можуть бути низькими, середніми або високими;
- за напрямком волатильності можуть бути зростаючими, падаючими або нейтральними.

Для вирішення аналітичних завдань існують різні типи машинного навчання: навчання без вчителя, навчання з вчителем, навчання з підкріпленням та напівавтоматичне [24]. У цій роботі розглядається тип машинного навчання, який називається навчанням з вчителем. Під навчанням з вчителем розуміється процес навчання моделі на наборі даних, який містить правильні відповіді.

Бінарна класифікація – це підтип навчання з вчителем, який використовується для розподілу об'єктів на два класи [24, 25].

Ось деякі конкретні переваги використання класифікації для обраної теми:

1) класифікація дозволяє розділити волатильності валют на групи, що мають спільні характеристики. Це може допомогти поліпшити розуміння волатильності валют, зробити її більш передбачуваною і допомогти у прийнятті рішень;

2) класифікація може використовуватися для розробки правил або моделей, які можна використовувати для прогнозування волатильності валют. Це може допомогти трейдерам і інвесторам приймати більш обґрунтовані рішення;

3) класифікація може використовуватися для управління валютним ризиком. Наприклад, можна використовувати класифікацію для визначення того, коли слід використовувати хеджування для захисту від ризику волатильності валют.

2.2 Методи класифікації машинного навчання

У сучасному світі, де обробка великого обсягу даних є необхідною частиною прийняття стратегічних рішень, методи класифікації машинного навчання виявляються надзвичайно важливими для аналізу та прогнозу різноманітних явищ.

Методи класифікації машинного навчання – це алгоритми та підходи, які дозволяють автоматично розподіляти об'єкти чи дані у визначені категорії або класи на основі їхніх характеристик чи ознак. Ці методи грають ключову роль у вирішенні задач класифікації та прогнозуванні в багатьох областях, включаючи медицину, фінанси, маркетинг, техніку та інші. Вибір методу класифікації залежить від конкретних цілей і даних, які використовуються.

Деякі з найпоширеніших методів класифікації машинного навчання описані нижче [26].

1. Наївний баєсівський класифікатор (Naive Bayes) – використовує теорему Баєса та припущення про незалежність ознак для класифікації об'єктів.

2. Метод опорних векторів (SVM) – шукає оптимальний гіперплощину для розділення об'єктів різних класів у високорозмірному просторі.

3. Дерева рішень (Decision Trees) – використовують структуру дерева для послідовного прийняття рішень та класифікації.

4. k-найближчих сусідів (K-Nearest Neighbors) – класифікує об'єкти на основі класів їхніх найближчих сусідів у вхідних даних.

5. Логістична регресія (Logistic Regression) – використовує логістичну функцію для моделювання ймовірностей класів.

6. Нейронні мережі (Neural Networks) – моделі, які імітують структуру та функції людського мозку для класифікації.

7. Градієнтний бустінг (Gradient Boosting) – комбінує декілька слабких моделей для покращення точності.

Це лише кілька прикладів, і вибір методу залежить від конкретного завдання, властивостей даних та вимог проекту.

В даному розділі ми розглядатимемо два популярних методи класифікації, а саме: наївний баєсівський класифікатор (NB) та метод опорних векторів (SVM). Ці методи виявляються ефективними у реалізації інтелектуальних систем. Розглянемо за допомогою таблиці 2.2 ключові аспекти, що роблять ці методи важливими та ефективними.

Таблиця 2.2 – Порівняльна характеристика наївного баєсівського класифікатора (NB) та методу опорних векторів (SVM)

Аспекти	**Наївний баєсівський класифікатор (NB)**	**Метод опорних векторів (SVM)**
Простота та швидкість навчання	Висока простота та швидкість, особливо при великих обсягах даних.	Ефективний, але трошки більше обчислювальної витрати.
Робота з великим обсягом даних	Легко масштабується для великих обсягів даних.	Добре працює з об'єктами в великорозмірному просторі ознак.

Продовження таблиці 2.2

Ефективність в текстовому аналізі	Застосовується з успіхом у задачах аналізу тексту.	Може бути використаний в аналізі тексту, але вимагає налаштувань.
Розділення класів в неоднорідних даних	Менш ефективний в умовах неоднорідних даних.	Добре працює з неоднорідними та складними даними.
Обмежений обсяг даних	Ефективний навіть при обмеженому обсязі даних.	Надає гарні результати в умовах обмежених ресурсів.

Ця таблиця відображає порівняльний аналіз наївного баєсівського класифікатора та методу опорних векторів за різними аспектами. Обидва методи мають свої сильні сторони та використовуються в різноманітних сценаріях залежно від вимог та особливостей задачі, тож розглянемо кожен з них більш детально.

Для класифікації волатильності валют за допомогою наївного баєсівського класифікатора та методу опорних векторів (SVM), можна використовувати різні ознаки, такі як процентний скачок, стандартне відхилення і інші метрики, які вказують на ступінь коливань цін.

2.2.1 Наївний баєсівський класифікатор (NB)

Наївний баєсівський класифікатор (Naive Bayes classifier) є алгоритмом машинного навчання, який базується на теоремі Баєса та вирішує проблеми класифікації [29, 30].

Алгоритм Баєса – є методом навчання з вчителем. Основна ідея полягає у визначенні ймовірності належності об'єкта до певного класу на основі ймовірностей входження кожної ознаки до цього класу. Назва "наївний" походить від припущення про незалежність між ознаками, що робить алгоритм вкрай простим, але дієвим.

У випадках, коли спостереження може належати до різних класів з різною ймовірністю, наївний баєсівський класифікатор видає результат у формі вектора, де кожна компонента представляє ймовірність належності об'єкта до певного класу.

Припускаючи, що кожен об'єкт описується n незалежними ознаками, класифікація відбувається за допомогою функції правдоподібності, яка розкладається на добуток ймовірностей для кожного класу. Сам класифікатор може бути параметричним або непараметричним, залежно від того, як одновимірні щільності відновлюються.

Цей підхід передбачає, що всі ознаки є взаємонезалежними, і результат класифікації визначається на підставі їхніх ймовірностей. Важливо враховувати це при використанні наївного баєсівського класифікатора, особливо у випадках, коли взаємозв'язки між ознаками складніші або коли припущення про їхню взаємну незалежність не виконується.

Основні етапи роботи наївного баєсівського класифікатора включають тренування та класифікацію. Під час тренування визначають ймовірності входження кожної ознаки для кожного класу, використовуючи навчальні дані. Після цього, під час класифікації для нового об'єкта обчислюють ймовірності його належності до кожного класу за допомогою теореми Баєса і обирають клас з найвищою ймовірністю.

Треба відзначити переваги наївного баєсівського класифікатора, серед яких низькі обчислювальні витрати під час навчання та класифікації, а також проста реалізація. У рідких випадках, коли ознаки є взаємонезалежними, наївний баєсів класифікатор може бути оптимальним вибором.

Зазначте, що важливо враховувати, що наївний баєсівський класифікатор передбачає взаємну незалежність ознак, що може бути ідеалізацією в реальних умовах. Також, цей метод ефективний в розпізнаванні патернів, але не завжди кращий для задач з складними взаємозв'язками між ознаками.

Цей класифікатор широко використовується в задачах, де кожна ознака може вносити незалежний внесок у визначення класу, таких як фільтрація спаму, аналіз тональності текстів, категоризація документів тощо.

Наприклад, можна використовувати наївний алгоритм Баєса для класифікації зображень тварин на основі їхніх рис. Алгоритм буде навчатися на наборі даних, який містить зображення тварин різних видів. Для кожного зображення алгоритм буде обчислювати ймовірність того, що воно належить до певного виду. Об'єкт буде класифікований у той вид, для якого ймовірність вища.

Алгоритм наївного Баєса ґрунтується на концепції ймовірностей, які використовуються для розв'язання невизначених завдань за допомогою передбачення [30]. Ймовірність визначає можливість настання події відносно всіх можливих результатів. Математично ймовірність події обчислюється як кількість сприятливих варіантів події поділена на загальну кількість можливих результатів, де ймовірність завжди лежить у межах від 0 до 1. Тут, 0 вказує на відсутність ймовірності події, а 1 – на абсолютну ймовірність успіху цієї події.

Наївний алгоритм Баєса також можна використовувати для класифікації текстових даних. Наприклад, алгоритм можна використовувати для класифікації електронних листів на спам і не спам. Алгоритм буде навчатися на наборі даних, який містить електронні листи різних класів. Для кожного електронного листа алгоритм буде обчислювати ймовірність того, що він належить до певного класу. Електронний лист буде класифікований у той клас, для якого ймовірність вища.

Формула Баєса використовується для обчислення ймовірності гіпотези H на основі нових даних або евіденції E [30, 31]. Теорія розглядає дві ключові ймовірності: апріорну ймовірність гіпотези до отримання доказів $P(H)$ і ймовірність гіпотези після отримання доказів $P(H|E)$. Це визначається співвідношенням:

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}, \quad (2.1)$$

де $P(H|E)$ – ймовірність гіпотези Н при наявності доказів Е;

$P(E|H)$ – ймовірність доказів Е при наявності гіпотези Н;

$P(H)$ – апріорна ймовірність гіпотези Н;

$P(E)$ – ймовірність свідчень Е.

Це рівняння Баєса дозволяє оновити ймовірність гіпотези на основі нових даних або доказів.

Умовна ймовірність визначає, наскільки ймовірно відбувається подія Y при умові, що подія X вже сталася [31]. Математично це виражається як $P(Y|X)$, що розраховується за формулою:

$$P(Y|X) = \frac{P(X \cap Y)}{P(X)}, \tag{2.2}$$

де $P(Y|X)$ – умовна ймовірність події Y при умові X;

$P(X \cap Y)$ – ймовірність спільності подій X та Y;

$P(X)$ – ймовірність події X.

Баєсова ймовірність відкриває можливість розрахунку умовних ймовірностей, дозволяючи використовувати часткові знання для визначення ймовірності настання конкретної події. Важливо враховувати, що точний вплив доказів на ймовірність події не завжди можна визначити, але можливо визначити ймовірність цієї події з урахуванням наявних даних.

Використовуючи умовну ймовірність, позначену як $P(E|O)$, можна обчислити ймовірність доказів на основі даних виходів. Задача полягає в обчисленні ймовірності результату $P(E|O)$, тобто ймовірності виходу на основі наданих атрибутів. У випадку, коли задача має два виходи, можна обчислити ймовірність для кожного результату та визначити, який з них ймовірніше відбудеться.

2.2.2 Метод опорних векторів (SVM)

Метод опорних векторів (англ. Support Vector Machine, SVM) є алгоритмом машинного навчання, який використовується для задач класифікації та регресії за допомогою набору алгоритмів навчання – опорно-векторних машин [33]. Модель використовує бінарний лінійний класифікатор для призначення елементів до категорій. Елементи представлені точками в просторі, таким чином, що елементи кожної категорії розташовані в окремій області площини, утвореній розділяючою площиною, яка має бути максимально широкою.

Метод SVM був розроблений у 1960-70-х роках та став особливо популярним у 90-ті роки XX століття [33]. Його навчання базується на вирішенні задачі квадратичного програмування, яка має єдиний розв'язок. Навіть при роботі з вибірками розміром у сотні тисяч об'єктів обчислення за допомогою цього методу залишається досить ефективним.

Основна ідея полягає в тому, щоб знайти оптимальну гіперплощину (в многовимірному просторі), яка найкращим чином розділяє об'єкти різних класів. Ця гіперплощина називається "оптимальною", оскільки вона має максимальний заздалегідь визначений зазор між класами.

Перелік основних термінів, пов'язаних із SVM наведено нижче [33, 34].

1) Гіперплощина – це (n-1)-мірна площина, де "n" – кількість ознак (або вимірів) у навчальних даних. У випадку класифікації, гіперплощина розділяє простір на дві частини, відповідні двом класам.

2) Оптимальний зазор – визначається як відстань від гіперплощини до найближчого об'єкта кожного класу. SVM шукає гіперплощину з максимальним зазором, щоб забезпечити найкращу універсальність моделі.

3) Опорні вектори – це точки навчальних даних, які лежать найближче до гіперплощини та визначають її положення. Ці вектори важливі для визначення гіперплощини та розділення класів.

4) Ядрові функції – SVM може використовувати ядрові функції, які дозволяють моделі розводити класи, якщо вони не є лінійно роздільними в оригінальному просторі. Ядрові функції дозволяють перетворювати дані у вищерозмірний простір, де вони можуть бути лінійно розділені.

SVM є потужним методом класифікації, особливо в ситуаціях з великою кількістю ознак чи неоднорідними даними. Однак для ефективної роботи він може вимагати правильного налаштування гіперпараметрів та досить обсяжний обсяг даних для навчання.

Рішення методу має різноманітні властивості, включаючи розрідженість, оскільки положення гіперплощини залежить від малої долі навчальних об'єктів. Ці об'єкти є опорними векторами, що й відповідно визначає назву методу. За допомогою введення функції ядра, яка є основним аспектом у методі, метод може бути узагальнений для нелінійних роздільних поверхонь. Однак вибір ядра є критичним моментом у методі: правильно підібране ядро дозволяє адаптувати метод для різних задач, не змінюючи його основу. Метод опорних векторів навіть дозволяє створювати прості двошарові нейронні мережі.

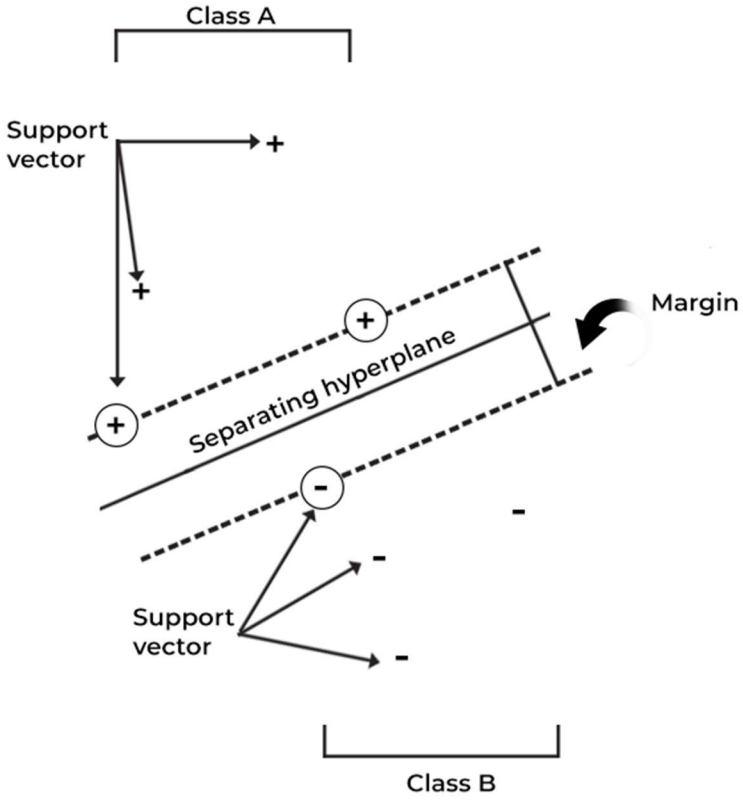


Рисунок 2.1 – Оптимізація відстані між опорними векторами або класами

Ядро SVM представляє собою симетричну, додатньо визначену матрицю K, що складається із скалярних добутків пар [35]:

$$x_i x_j : K(x_i, x_j) = < f(x_i), f(x_j) >, \quad (2.3)$$

де f – довільна перетворююча функція для формування ядра.

Наприклад:

1) лінійне ядро визначається як скалярний добуток між векторами ознак x_i та x_j: $K(x_i, x_j) = x_i^T x_j$;

2) сигмоїдне ядро визначається за допомогою тангенсу з аргументом, який включає лінійну комбінацію між векторами ознак x_i та x_j, а також параметрами γ та β_0: $K(x_i, x_j) = tan(\gamma x_i^T x_j + \beta_0)$;

3) гаусове ядро з радіальною базовою функцією описується експонентою від квадрату відстані між векторами ознак x_i та x_j, множеною на параметр γ: $K(x_i, x_j) = exp(\gamma\|x_i - x_j\|^2)$;

4) поліноміальне ядро зі степенем p визначається за допомогою виразу $(1 + x_i^T x_j)$ в піднятому до ступеня p: $K(x_i, x_j) = (1 + x_i^T x_j)^p$.

Відстань від векторів до гіперплощини в контексті методу опорних векторів називається розділенням або краєм (англ. margin). Це поняття визначає відокремлення, або зазор, між гіперплощиною, яка служить розділювальною границею між класами, і найближчими точками даних цих класів.

У SVM при лінійному відокремленні даних вибирається гіперплощина таким чином, щоб цей зазор був максимальним. Зазор вимірюється відстанню від кожної точки до гіперплощини.

Для кращого розуміння, існує два види розділення [34, 35]:

1) жорстке розділення (Hard Margin) – вимагає, щоб всі точки одного класу знаходились по один бік від гіперплощини, і зазор був максимальним. Це може призвести до неправильної класифікації точок, якщо дані не є лінійно відокремлюваними;

2) м’яке розділення (Soft Margin) – дозволяє деяким точкам потрапляти на інший бік гіперплощини. Це важливо для справжніх (нелінійно відокремлюваних) даних, де жорстке розділення може призвести до неправильної класифікації. Зазор при цьому стає компромісом між максимізацією та допустимим порушенням класифікації.

Якщо тренувальна вибірка є лінійно роздільною, метод опорних векторів (SVM) може вибрати дві паралельні гіперплощини для класифікації даних так, щоб відстань між ними була максимальною. Враховуючи це, розглянемо простий випадок класифікації методом SVM (рис. 2.2):

- L1 – гіперплощина, яка не відповідає умовам SVM і не правильно розділяє дані;
- L2 – гіперплощина, яка правильно розділяє дані на дві множини, але не є максимальною роздільною площиною, оскільки є більший зазор, який можна використовувати;
- L3 – гіперплощина, яка задовольняє усім умовам SVM і є максимальним розділенням. Вона розташована так, щоб максимізувати відстань між класами, забезпечуючи ефективну роздільну межу.

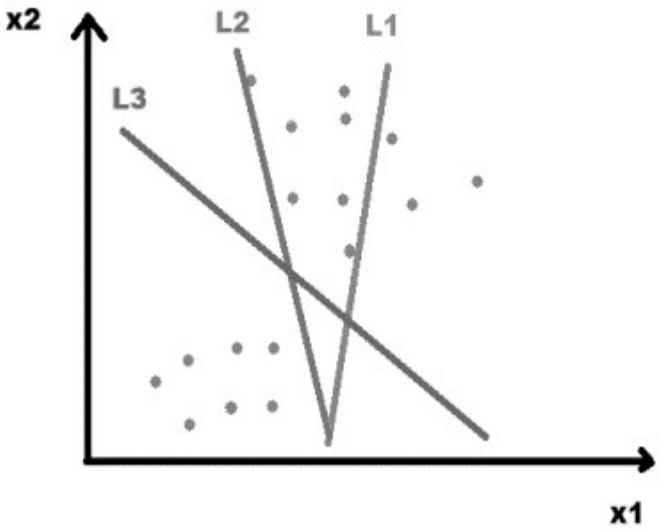


Рисунок 2.2 – Метод опорних векторів (SVM)

Отже, метод SVM вибирає гіперплощину (позначену як L3), яка найкращим чином розділяє дані та максимізує зазор між класами, що робить його ефективним для класифікації в лінійно роздільних випадках.

SVM використовує геометричні властивості для визначення відділяючої гіперплощини навіть з тренувальних даних. Цей метод дозволяє взаємодіяти з нелінійними розділяючими елементами, таким чином враховуючи ситуації, коли

рішенням є не пряма лінія. Він також відмінно справляється із завданнями, що містять шум у тренувальних даних, де присутні неправильні елементи у вибірці.

Класифікатор SVM включає три основні параметри для навчання: ядро, параметр регуляризації та параметр гамма. Ми вже розглядали поняття ядра раніше.

Параметр регуляризації (зазвичай позначається як C) визначає точність класифікації в методі опорних векторів. При великих значеннях C оптимізація обере гіперплощину з меншим полем, якщо ця гіперплощина краще відображає класифікацію тренувальних даних. Навпаки, дуже мале значення C змусить оптимізатор шукати гіперплощину, що розділяє більші області, навіть якщо вона неправильно класифікує більше даних.

Параметр гамма визначає, наскільки вагомим є вплив одного прикладу тренувань, де низькі значення означають "малий", а високі значення означають "великий". Іншими словами, при низькій гаммі точки, віддалені від правдоподібних ліній розділу, враховуються при розрахунку для лінії розділу, тоді як велика гамма означає, що точки, близькі до правдоподібних ліній, враховуються при розрахунку.

Існують інші гіперпараметри, які відіграють важливу роль, проте деякі з них є специфічними для задачі класифікації або регресії, та залежать від інших гіперпараметрів. Наприклад, параметр "степінь" використовується лише для поліноміального ядра, інші ж ігнорують його.

2.3 Метрики оцінки якості роботи класифікаторів

До цього часу точність класифікатора визначалася як відношення кількості правильних прогнозів до загальної кількості прогнозів. Тобто ми отримували частку випадків, коли алгоритм навчання був правильний. Проте перед тим, як використовувати класифікатор на практиці для прогнозування, розумно збирати додаткову інформацію щодо його ефективності. При оцінці класифікатора важливо враховувати проблему дисбалансу класів, яка виникає, коли більшість записів у наборі

даних відноситься до одного класу. Це є актуальною проблемою і в нашому досліджуваному наборі даних.

Існує безліч способів вимірювання ефективності класифікатора, але найбільш об'єктивним є той, який визначає, наскільки класифікатор успішний у реальних ситуаціях за його прямим призначенням. Важливо враховувати показники ефективності, які відображають корисність, а не лише точність. Для отримання альтернативних показників ефективності ми можемо використовувати матрицю невідповідностей.

Правильні класифікації відзначаються, коли прогнозовані та реальні значення збігаються, і такі випадки відображаються на діагоналі матриці невідповідностей. З іншого боку, неправильні прогнози відображаються в комірках матриці, які знаходяться поза діагоналлю. Показники ефективності для моделей класифікації ґрунтуються на кількості прогнозів, які входять до діагоналі та за її межі.

Найбільш загальні показники ефективності враховують здатність моделі відрізняти один клас від іншого. У цьому контексті цікавий клас визначається як позитивний, а решта – як негативний.

Матриця невідповідностей 2x2, також відома як таблиця сплутаності, є інструментом для візуалізації залежності між прогнозами позитивного та негативного класу. У цій матриці представлено чотири можливі категорії для прогнозів [37].

Представимо, що розглядається задача оцінки волатильності валют, де y = 1 вказує на високий рівень волатильності, а y = 0 – на низький рівень волатильності.

1) True Positive (TP) – кількість правильно визначених позитивних прикладів.

2) False Positive (FP) – кількість неправильно визначених позитивних прикладів.

3) True Negative (TN) – кількість правильно визначених негативних прикладів.

4) False Negative (FN) – кількість неправильно визначених негативних прикладів.

Коефіцієнт помилок (error rate), або частка невірно класифікованих прикладів визначається так (рис. 2.3):

Predicted values	True Values	y = 1	y=0
	a(x) = 1	True Positive (TP)	False Positive (FP)
	a(x) = 0	False Negative (FN)	True Negative (TN)

Рисунок 2.3 – Матриця невідповідностей

За допомогою цієї матриці можна розрахувати різні метрики ефективності класифікатора, такі як чутливість (recall), специфічність (specificity), точність (precision), F1-міра і інші. Ці метрики надають детальну інформацію про роботу класифікатора в різних аспектах.

Не секрет, що хотілося би того, щоб наша інформаційна система видавала завжди вірний безпомилковий результат, але у реальному житті таке маловірогідне, тому кожна модель повинна мінімізувати кількість хибних результатів. Вибір найкращої метрики для оцінки ефективності моделі машинного навчання залежить від конкретних потреб і цілей [37].

Точність (Accuracy) – це метрика, яка визначає загальну ефективність класифікатора і вимірює відсоток правильно класифікованих випадків у відношенні до загальної кількості випадків [38]. Вона розраховується за формулою:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2.4)$$

Наприклад, якщо модель машинного навчання зробила 100 прогнозів і 80 з них були правильними, то точність моделі дорівнює 0,8, або 80%.

Точність є загальною метрикою, яка може використовуватися для оцінки ефективності моделей машинного навчання в різних задачах. Однак, вона може бути обманливою в умовах дисбалансу класів, коли один клас переважає над іншим.

Наприклад, точність може бути не такою важливою, як чутливість або специфічність, якщо модель використовується для виявлення рідкісних подій.

Чіткість (Precision) – це метрика, яка визначає відсоток елементів, які видає класифікатор як позитивні, що дійсно є позитивними [38]. Ця метрика дозволяє оцінити, наскільки впевнено класифікатор визначає позитивні приклади. Формула для обчислення чіткості така:

$$Precision = \frac{TP}{TP + FP}, \quad (2.5)$$

Чим вище чіткість, тим менше можливість отримати "хибний позитив" (неправильно визначений позитивний приклад).

Наприклад, якщо модель машинного навчання зробила 100 прогнозів, з яких 20 були позитивними, і 18 з цих 20 позитивних прогнозів були правильними, то чіткість моделі дорівнює 0,9, або 90%.

Чіткість важлива в таких випадках, коли важливо уникати помилкових позитивних прогнозів. Наприклад, якщо модель машинного навчання використовується для виявлення злочинів, то важливо, щоб модель не класифікувала випадкову подію як злочин. В іншому випадку, це може призвести до арешту невинної людини.

Повнота (Recall) – це метрика, яка вимірює, як часто модель машинного навчання правильно класифікує негативні приклади [38]. Вона визначається як кількість правильних негативних прогнозів, поділена на загальну кількість негативних прикладів.

Повнота є важливою метрикою для оцінки ефективності моделей машинного навчання, які використовуються для виявлення поширених подій. Наприклад, якщо

модель машинного навчання використовується для виявлення спаму, повнота визначає, як часто модель правильно класифікує спамову пошту.

Повноту можна розрахувати за такою формулою:

$$Recall = \frac{TP}{TP + FN}, \tag{2.6}$$

Наприклад, якщо модель машинного навчання зробила 100 прогнозів, з яких 80 були негативними, і 72 з цих 80 негативних прогнозів були правильними, то повнота моделі дорівнює 0,9, або 90%.

F-міра – це міра точності, яка враховує як точність, так і повноту. Вона визначається як гармонійне середнє цих двох метрик [38].

F-міра може бути розрахована за такою формулою:

$$F = \frac{2 * precision * recall}{precision + recall}, \tag{2.7}$$

де precision – це кількість правильних позитивних прогнозів, поділена на загальну кількість позитивних прогнозів;

recall – це кількість правильних негативних прогнозів, поділена на загальну кількість негативних прогнозів.

F-міра є важливою метрикою для оцінки ефективності моделей машинного навчання, які використовуються для виявлення рідкісних подій. У таких випадках важливо, щоб модель мала високу точність, щоб уникати помилкових позитивних прогнозів, але також важливо, щоб вона мала високу повноту, щоб не пропускати справжні позитивні випадки.

Висновки до розділу 2

В результаті даного розділу кваліфікаційної роботи магістра було проведено вивчення та аналіз вибору методів машинного навчання для вирішення задачі класифікації.

Визначено та розглянуто основні поняття класифікації як одного з ключових напрямків машинного навчання, та різні методи машинного навчання, зокрема наївний баєсівський класифікатор (NB) та метод опорних векторів (SVM). Розкрито важливість правильного вибору методів для ефективного розв'язання задачі класифікації.

Також було детально проаналізовано різні метрики оцінки якості роботи класифікаторів. Вибір правильних метрик є важливим етапом у визначенні ефективності моделей класифікації, і проведений аналіз допоможе враховувати різні аспекти роботи класифікатора.

Отримані знання про класифікацію, обрані методи та метрики надають підґрунтя для подальших досліджень у розв'язанні та реалізації конкретної задачі машинного навчання.

3 МОДЕЛЮВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЛАСИФІКАЦІЇ ВОЛАТИЛЬНОСТІ ВАЛЮТ

3.1 Програмні засоби для розробки інтелектуальної системи

Розробка інтелектуальних систем, зокрема системи класифікації волатильності валют, вимагає ретельного підходу до вибору технологій, які будуть використовуватися під час процесу створення програмного забезпечення.

Одним із ключових аспектів при виборі технологій для розробки інтелектуальної системи є мова програмування, яка дозволяє зручно і ефективно втілювати в життя поставлені завдання. У даному випадку, вибір падає на мову програмування Python з ряду вагомих причин.

Python – це високорівнева, інтерпретована, загальнопризначена мова програмування з простим та читабельним синтаксисом, широким спектром застосувань, великою активною спільнотою розробників та розширеною екосистемою бібліотек і фреймворків. Крім цього, це мова програмування, що є платформо незалежною, і, завдяки цьому, може бути легко адаптована для використання на різних операційних системах. Він був створений Python Foundation на початку 1990-х і є потужним інструментом для аналізу даних, що широко використовується в технології великих даних.

Рисунок 3.1 – Логотип мови програмування Python

Вибір Python для реалізації інтелектуальної системи мотивований його використанням в промисловості та наукових дослідженнях, зокрема у сферах машинного навчання та штучного інтелекту. Велика кількість корпорацій та наукових установ, таких як Google, Facebook, NASA, вибрали Python для своїх проектів. Він відзначається своєю простотою в освоєнні, що дозволяє розробникам швидко реалізовувати та тестувати ідеї.

Розглядаючи впровадження машинного навчання можна встановити, що Python є однією з провідних мов для цього напрямку завдяки бібліотекам, таким як TensorFlow, PyTorch, Scikit-learn та інших.

Обрання Python для створення телеграм-боту також має обґрунтовані підстави. Дана мова програмування має потужні бібліотеки, наприклад, Telebot, які спрощують та полегшують процес розробки ботів. Це дозволяє створювати інтерактивні та динамічні інтерфейси для взаємодії з користувачем, отримуючи та надсилаючи дані зручним способом.

Тож розглянемо ключові бібліотеки, які використовувалися при розробці інтелектуальної системи для аналізу та класифікації волатильності валют:

1) Pandas та NumPy – для ефективного завантаження, обробки та маніпуляції даними були використані бібліотеки Pandas та NumPy. Вони надають зручний інтерфейс для роботи з табличними даними та векторами;

2) Scikit-learn – бібліотека scikit-learn стала основним інструментом для реалізації методів машинного навчання та класифікації. Вона включає широкий спектр алгоритмів, а також інструменти для підготовки даних та оцінки моделей;

3) Matplotlib та Seaborn – для візуалізації даних та результатів були використані бібліотеки Matplotlib та Seaborn. Вони надають зручні засоби для побудови графіків та діаграм, що сприяють кращому розумінню результатів аналізу;

4) Requests – для здійснення API-запитів та отримання актуальних даних про валютні курси була використана бібліотека requests. Вона забезпечує зручний інтерфейс для взаємодії зі зовнішніми веб-ресурсами;

5) Telebot – для створення та управління телеграм-ботом використовувалася бібліотека Telebot, яка дозволяє реалізувати зручний та інтерактивний інтерфейс для користувачів.

Ці бібліотеки допомогли створити комплексну систему для аналізу та класифікації волатильності валют, обробки даних, реалізації алгоритмів машинного навчання та взаємодії з користувачами через телеграм-бот.

Крім цього, не можна не зазначити, що впровадження Django у процес розробки телеграм-бота стало ключовим кроком для досягнення мети створення інтелектуальної системи з класифікації волатильності валют. Django, як високорівневий веб-фреймворк, пропонує безліч переваг, які значно полегшують роботу розробників та роблять проект більш масштабованим і гнучким.

Його використання дозволяє нам не лише забезпечити ефективну роботу Telegram-бота, але й створити додатковий веб-інтерфейс для адміністрування та керування системою. Адміністративний інтерфейс Django виявляється вельми зручним інструментом для взаємодії з конфігурацією та базою даних, що є ключовими складовими нашого проекту.

Незважаючи на те, що Python є однією з найпопулярніших мов програмування для машинного навчання, вона також має свої недоліки. Декілька з них описані нижче.

1. Швидкодія. В порівнянні з іншими мовами, такими як C++ або Java, Python може бути менш ефективним у виконанні обчислювально важких завдань через інтерпретацію коду. Це може вплинути на швидкодію великих обчислювальних завдань у машинному навчанні.

2. Менше підтримки для розподілених обчислень. Деякі інші мови програмування, такі як Scala чи Java, мають більше підтримки для розподілених обчислень, що може бути важливим для роботи з великими обсягами даних.

3. Обмежена підтримка для розробки мобільних додатків. Якщо потрібно вбудувати модель машинного навчання безпосередньо в мобільний додаток, інші мови, такі як Swift для iOS чи Kotlin для Android, можуть бути більш підходящими.

4. Обмежені можливості для обробки графіки. Якщо ваше завдання включає великий обсяг роботи з графікою, Python може виявитися менш ефективним, оскільки він не так сильно орієнтований на роботу з графікою як, наприклад, R чи Matlab.

Реалізація проекту проводилася в інтегрованому середовищі розробки PyCharm (рис. 3.2), яке є одним із найпопулярніших та потужних інструментів для розробки на мові програмування Python. PyCharm надає розробникам ряд важливих можливостей та інструментів, що полегшують процес кодування та підтримують збірку великих та складних проектів.

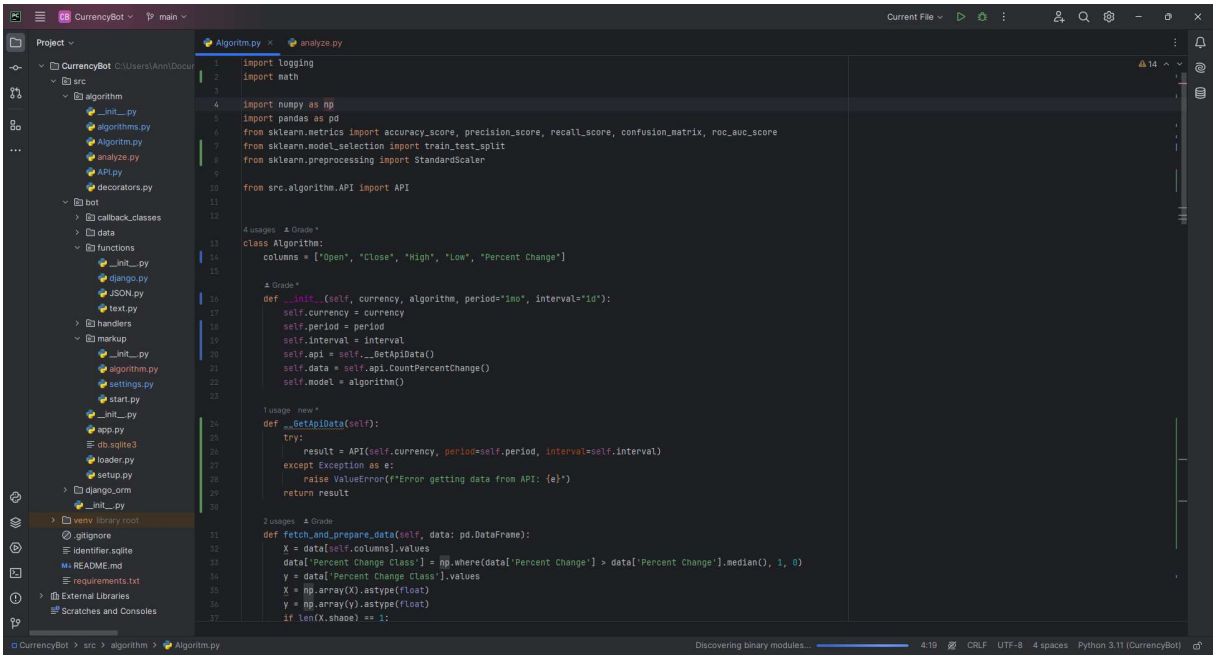


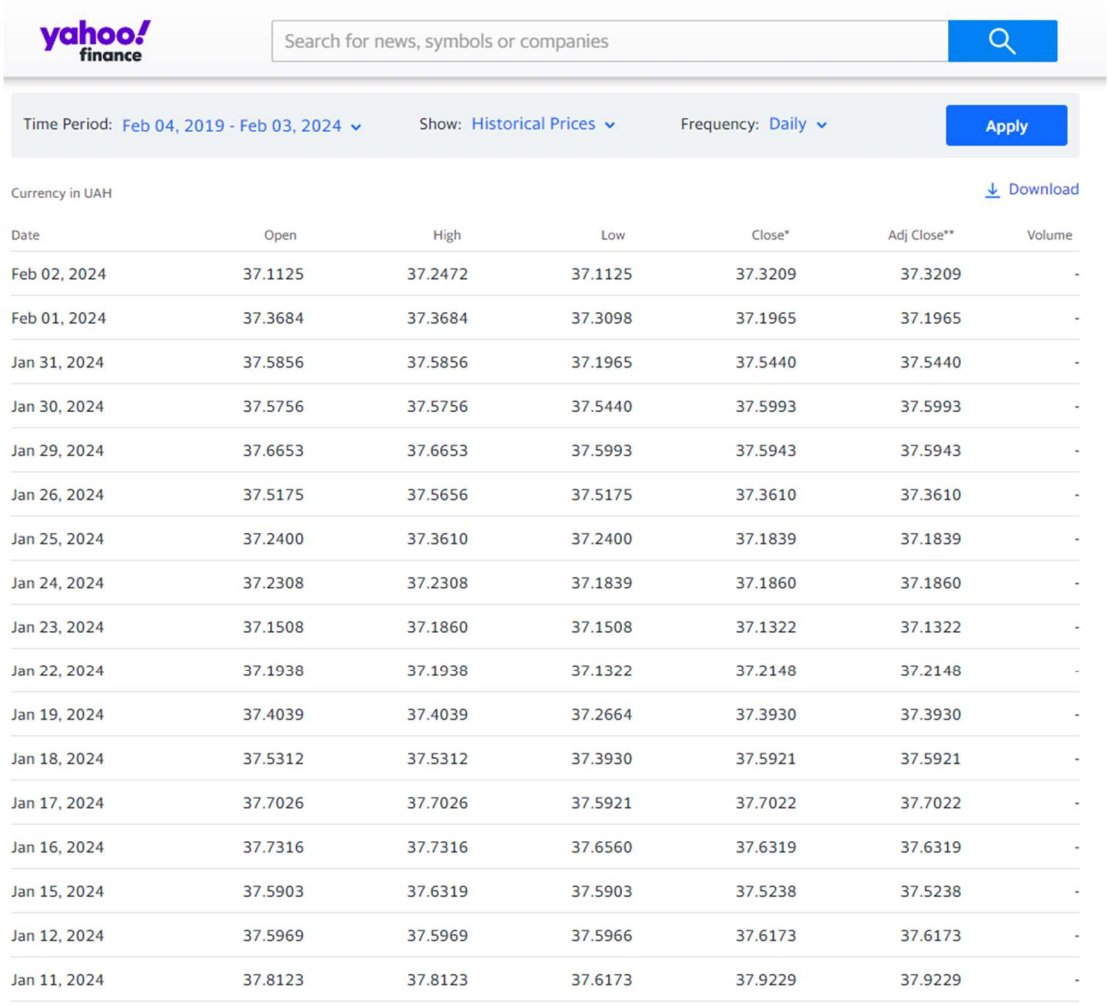
Рисунок 3.2 – Інтерфейс PyCharm

Його інтегровані функції включають автоматичне завершення коду, рефакторинг, інструменти аналізу коду, систему керування версіями, і багато іншого. PyCharm допомагає підтримувати чистоту та структурованість коду, що важливо для ефективної та швидкої розробки проекту.

Його інтуїтивний інтерфейс та можливість інтеграції з різними інструментами створюють зручне середовище для розробки та тестування інтелектуальних систем, таких як система класифікації волатильності валют, що нам передбачено в цьому проекті.

3.2 Загальний опис набору даних

Для наглядного прикладу проведення аналізу та попередньої обробки був використаний ключовий набір даних USD/UAH, отриманий за допомогою Yahoo Finance API. На основі цього набору даних ми провели докладний аналіз, спрямований на визначення важливих параметрів та особливостей. Головною задачею наших подальших моделей є оцінка рівня волатильності валютної пари USD/UAH та її класифікація на низький чи високий рівень. Обраний датасет відображає динаміку валютного курсу долара США до української гривні протягом останніх п’яти років, з 4 лютого 2019 року по 3 лютого 2024 року.



Date	Open	High	Low	Close*	Adj Close**	Volume
Feb 02, 2024	37.1125	37.2472	37.1125	37.3209	37.3209	-
Feb 01, 2024	37.3684	37.3684	37.3098	37.1965	37.1965	-
Jan 31, 2024	37.5856	37.5856	37.1965	37.5440	37.5440	-
Jan 30, 2024	37.5756	37.5756	37.5440	37.5993	37.5993	-
Jan 29, 2024	37.6653	37.6653	37.5993	37.5943	37.5943	-
Jan 26, 2024	37.5175	37.5656	37.5175	37.3610	37.3610	-
Jan 25, 2024	37.2400	37.3610	37.2400	37.1839	37.1839	-
Jan 24, 2024	37.2308	37.2308	37.1839	37.1860	37.1860	-
Jan 23, 2024	37.1508	37.1860	37.1508	37.1322	37.1322	-
Jan 22, 2024	37.1938	37.1938	37.1322	37.2148	37.2148	-
Jan 19, 2024	37.4039	37.4039	37.2664	37.3930	37.3930	-
Jan 18, 2024	37.5312	37.5312	37.3930	37.5921	37.5921	-
Jan 17, 2024	37.7026	37.7026	37.5921	37.7022	37.7022	-
Jan 16, 2024	37.7316	37.7316	37.6560	37.6319	37.6319	-
Jan 15, 2024	37.5903	37.6319	37.5903	37.5238	37.5238	-
Jan 12, 2024	37.5969	37.5969	37.5966	37.6173	37.6173	-
Jan 11, 2024	37.8123	37.8123	37.6173	37.9229	37.9229	-

Рисунок 3.3 – Набір даних USD/UAH

Одним із ключових показників для класифікації волатильності є зміна цін валют протягом певного періоду часу. Цей показник взятий за основу для аналізу та визначення рівня ризику. Ми враховуємо історичні дані про денні чи інші часові зміни курсів валют, обсяги торгів та інші фактори, які можуть впливати на волатильність.

Детальний аналіз зміни цін валют допомагає впевнено класифікувати валютні пари на рівні низької чи високої волатильності, що в свою чергу сприяє кращому управлінню ризиками та прийняттю обґрунтованих рішень в області фінансових операцій. Низький рівень волатильності може вказувати на стабільність та

прогнозованість, тоді як високий рівень може свідчити про нестабільність та більші коливання курсу.

Розглядаючи ці аспекти, система класифікації спроможна надавати важливі інсайти щодо ризиків на валютному ринку, допомагаючи фінансовим аналітикам та інвесторам приймати обґрунтовані рішення в умовах невизначеності та змін на фінансовому ринку.

В поданому датасеті для валют ми маємо наступні ключові атрибути, що істотно впливають на аналіз динаміки ринку:

1) дата (Date) – параметр визначає точний момент часу вимірювань. Він виступає як хронологічний відлік для всіх подальших атрибутів та є визначальним для аналізу валютних подій;

2) курс відкриття (Open) – значення відображає курс валюти на момент відкриття торгів. Визначає першочерговий інтерес до активів та устанавлює початковий рівень цінової активності;

3) максимальний курс (High) – атрибут вказує на максимальний зафіксований курс валюти протягом періоду торгів. Визначає пікове значення цінових коливань;

4) мінімальний курс (Low) – цей параметр визначає найнижчий зафіксований курс валюти протягом торгового періоду. Реєструє мінімальне значення цін;

5) курс закриття (Close) – вказує на курс валюти на момент закриття торгів. Це фінальне значення, яке може відобразити реальний вплив факторів на курс;

6) скоригований закритий курс (Adj Close) – скориговане значення закриття, що враховує додаткові фактори, такі як поділ акцій або інші коригуючі події. Використовується для точнішого аналізу;

7) обсяг торгів (Volume) – вказує на кількість валюти, якою укладено угоди протягом вказаного періоду. Цей атрибут служить індикатором ринкової активності та об'єму торгів;

Використання цих атрибутів надасть нам можливість детально вивчати та розуміти динаміку цін на валюту, а також аналізувати тенденції та ринкову активність.

Проте в рамках подальшого аналізу цього датасету ми плануємо обчислити додаткові показники, які дозволять нам кількісно оцінити рівень волатильності валют, який в подальшому буде використовуватися для класифікації валютних пар на низький або високий рівень.

3.3 Попередня обробка даних. Підготовка до моделювання

Попередня обробка даних – це процес підготовки та очищення даних перед їх використанням у моделях машинного навчання. Як правило, при обробці ми маємо справу з великими обсягами необроблених вихідних даних. Алгоритми машинного навчання розраховані на те, що, перш ніж вони зможуть розпочати процес тренування, отримані дані будуть відформатовані певним чином. Щоб привести дані до форми, що прийнятна для алгоритмів машинного навчання, ми повинні попередньо підготувати їх і перетворити на потрібний формат. Покажемо, як це робиться.

Функція read_csv() є частиною бібліотеки Pandas у мові програмування Python і призначена для завантаження даних з файлів у форматі CSV (Comma-Separated Values). Формат CSV використовується для збереження табличних даних у текстовому файлі, де значення розділені комами або іншим роздільником. Ви повинні вказати повний шлях до файлу у рядку, переданим функції.

```
import pandas as pd
df = pd.read_csv(r'C:\Users\Ann\Documents\Магістерська\UAH=X.csv')
```

Після виконання цього коду змінна df буде містити кадр даних з інформацією з CSV-файлу. У нашому випадку це дані валютної пари USD/UAH за період останніх п’яти років. Набір даних містить 1305 спостережень.

Такий підхід також можна використовувати для завантаження даних з інших форматів файлів, таких як Excel, SQL, JSON та інші, залежно від потреб та формату ваших даних.

У Python для отримання назв стовпців (змінних) використовується метод .columns для об'єкта DataFrame бібліотеки pandas.

```
column_names = df.columns
print(column_names)
```

```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

Рисунок 3.4 – Отримання назв стовпців

Для перегляду перших і останніх рядків з набору даних, використаємо метод .head() та .tail().

```
# Перегляньте перші рядки даних
print(df.head())
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-02-04	27.302999	27.315001	27.136000	27.288000	27.288000	0
1	2019-02-05	27.177000	27.315001	26.917999	27.148001	27.148001	0
2	2019-02-06	26.955999	27.232000	26.955999	26.959000	26.959000	0
3	2019-02-07	26.760000	26.969999	26.760000	26.969999	26.969999	0
4	2019-02-08	26.674000	26.969999	26.674000	26.969999	26.969999	0

Рисунок 3.5 – Перегляд перших рядків з набору данних

```
# Перегляньте останні рядки даних
print(df.tail())
```

	Date	Open	High	Low	Close	Adj Close
1300	2024-01-30	37.575554	37.575554	37.543957	37.599323	37.599323
1301	2024-01-31	37.585552	37.585552	37.196495	37.543957	37.543957
1302	2024-02-01	37.368393	37.368393	37.309834	37.196495	37.196495
1303	2024-02-02	37.112518	37.247242	37.112518	37.320877	37.320877
1304	2024-02-03	37.562698	37.604698	37.562698	37.604698	37.604698

Рисунок 3.6 – Перегляд останніх рядків з набору данних

Виведення структури набору даних для подальшого аналізу виконується за допомогою функції info().

```
print(df.info()
```

```
RangeIndex: 1305 entries, 0 to 1304
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       1305 non-null   object
 1   Open       1305 non-null   float64
 2   High       1305 non-null   float64
 3   Low        1305 non-null   float64
 4   Close      1305 non-null   float64
 5   Adj Close  1305 non-null   float64
 6   Volume     1305 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 71.5+ KB
None
```

Рисунок 3.7 – Виведення структури набору даних

Факторні або нечислові змінні в статистиці і аналізі даних – це змінні, які приймають значення з обмеженого та конкретного набору категорій або рівнів. Факторні змінні включають "Date" (Дата), оскільки це категоріальний об'єкт. Числові

змінні включають "Open" (Відкриття), "High" (Максимальна), "Low" (Мінімальна), "Close" (Закриття), "Adj Close" (Скориговане закриття), "Volume" (Обсяг).

Нечислові та відсутні значення в наборі даних перевіряються за допомогою методу isnull().

```
missing_values = df.isnull()
print(missing_values)
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
1298	False	False	False	False	False	False	False
1299	False	False	False	False	False	False	False
1300	False	False	False	False	False	False	False
1301	False	False	False	False	False	False	False
1302	False	False	False	False	False	False	False

Рисунок 3.8 – Перевірка на нечислові та відсутні значення

При перевірці з’ясовано, що в обраному для досліджень наборі даних немає нечислових чи відсутніх значень в жодному атрибуті.

У Python для отримання узагальненої інформації про набір даних, ви можете використовувати метод .describe() для об'єкта DataFrame бібліотеки pandas. Цей метод надає основні статистичні параметри для кожного стовпця даних.

```
# Виведіть узагальнену інформацію про набір даних
print(df.describe())
```

Цей код виведе основні статистичні параметри для числових стовпців, такі як: count – кількість, mean – середнє значення, std – стандартне відхилення, min -

мінімальне та max – максимальне значення. 25%, 50%, і 75% – це квартилі в статистиці, які визначають крайні значення в певному відсотковому розподілі даних. Це допоможе вам отримати узагальнену інформацію про розподіл даних та їх характеристики.

	Open	High	Low	Close	Adj Close	Volume
count	1305.000000	1305.000000	1305.000000	1305.000000	1305.000000	1305.0
mean	29.816734	29.882907	29.762852	29.817748	29.817748	0.0
std	4.711851	4.722967	4.702306	4.704829	4.704829	0.0
min	22.922478	22.946453	22.922478	22.946453	22.946453	0.0
25%	26.493023	26.553030	26.445000	26.482706	26.482706	0.0
50%	27.675570	27.703550	27.634747	27.682863	27.682863	0.0
75%	36.353676	36.485634	36.328556	36.446461	36.446461	0.0
max	37.937950	37.956085	37.878925	37.989044	37.989044	0.0

Рисунок 3.8 – Виведення узагальненої інформації про набір даних

Проаналізувавши результати виведення .describe() для набору даних, можна зробити наступні висновки:

1) кількість (count): усі числові стовпці (Open, High, Low, Close, Adj Close, Volume) мають однакову кількість значень, що дорівнює 1303. Це вказує на відсутність пропущених значень у цих стовпцях;

2) середнє значення (mean): середні значення числових стовпців вказують на середні ціни та обсяг торгів під час вказаного періоду;

3) стандартне відхилення (std): стандартне відхилення вимірює розкид значень від середнього. Малі значення стандартного відхилення вказують на те, що дані тенденційні та близькі одне до одного;

4) мінімальне та максимальне значення (min, max): мінімальні та максимальні значення вказують на екстремальні точки у розподілі. У вас є інформація про найнижчі та найвищі ціни під час вказаного періоду;

5) квартилі (25%, 50%, 75%): квартилі надають інформацію про розподіл значень. Наприклад, 25-й квартиль (Q1) вказує на те, що 25% значень менше або рівні цьому значенню, а 75-й квартиль (Q3) вказує на те, що 75% значень менше або рівні цьому значенню.

Ці висновки дозволяють зробити загальний огляд числових характеристик набору даних і виявити основні тенденції та варіації у ваших даних.

Провівши додатковий аналіз, видаляються ознаки, що не мають безпосереднього впливу на подальшу класифікацію. Щоб видалити кілька стовпців (у нашому випадку 'Volume' і 'Adj Close') з датафрейму, ви можете передати список імен стовпців до методу drop.

```
# Видаліть стовпці 'Volume' і 'Adj Close'
columns_to_drop = ['Volume', 'Adj Close']
df = df.drop(columns=columns_to_drop, axis=1)
# Виведіть оновлений датафрейм
print(df.head())
```

	Date	Open	High	Low	Close
0	2019-02-04	27.302999	27.315001	27.136000	27.288000
1	2019-02-05	27.177000	27.315001	26.917999	27.148001
2	2019-02-06	26.955999	27.232000	26.955999	26.959000
3	2019-02-07	26.760000	26.969999	26.760000	26.969999
4	2019-02-08	26.674000	26.969999	26.674000	26.969999

Рисунок 3.9 – Видалення ознак та виведення оновленого набору даних

В результаті відбору ознак набір даних матиме таку структуру:

```
RangeIndex: 1305 entries, 0 to 1304
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    1305 non-null   object
 1   Open    1305 non-null   float64
 2   High    1305 non-null   float64
 3   Low     1305 non-null   float64
 4   Close   1305 non-null   float64
dtypes: float64(4), object(1)
memory usage: 51.1+ KB
None
```

Рисунок 3.10 – Виведення оновленої структури набору даних

Наступним етапом є побудова діаграм розподілу (рис. 3.11). Діаграми розподілу – це графічний спосіб візуалізації розподілу даних. Ці діаграми дозволяють побачити, як розподілені значення змінної по всьому її діапазону. Діаграми розподілу надають інформацію про форму розподілу, центральні тенденції, розкид та можливі викиди або аномалії в даних.Функція автоматично конвертує логічний та факторний тип даних в числовий, для відображення даних в вигляді діаграм.

```
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns

for column in numeric_columns:
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True, color='blue')
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()
```

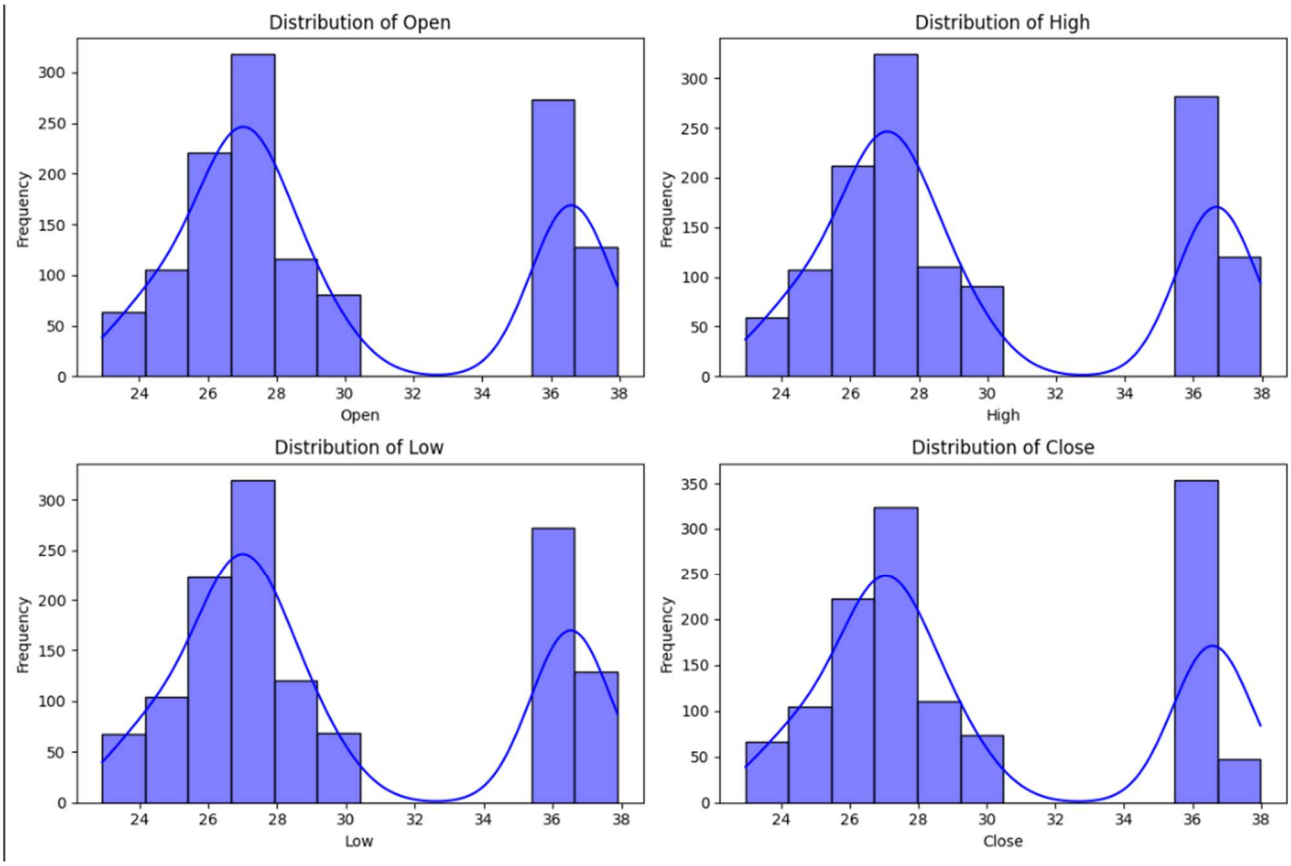


Рисунок 3.11 – Діаграми розподілу для числових стовпців

Для класифікації волатильності валют можна обрати різні ознаки, які допоможуть системі визначати та присвоювати рівні волатильності. Проте декілька виділених мною ознак, таких як процентне змінення та стандартне відхилення, можуть бути особливо значущими. Давайте розглянемо їх більш детально.

Процентне змінення (Percent Change). Ця ознака визначає, наскільки змінилася ціна валюти від відкриття до закриття. Вона розраховується як відсоткова різниця між ціною закриття та ціною відкриття.

Високе значення цієї ознаки може свідчити про значний рух цін протягом періоду, що може бути індикатором волатильності.

```
df['Percent Change'] = (df['Close'] - df['Open']) / df['Open'] * 100
```

None

	Date	Open	High	Low	Close	Percent Change
0	2019-02-04	27.302999	27.315001	27.136000	27.288000	-0.054935
1	2019-02-05	27.177000	27.315001	26.917999	27.148001	-0.106704
2	2019-02-06	26.955999	27.232000	26.955999	26.959000	0.011133
3	2019-02-07	26.760000	26.969999	26.760000	26.969999	0.784750
4	2019-02-08	26.674000	26.969999	26.674000	26.969999	1.109691
5	2019-02-11	26.851000	26.865999	26.754000	26.825001	-0.096827
6	2019-02-12	26.837999	26.865999	26.802999	26.865999	0.104330
7	2019-02-13	26.700001	26.865999	26.700001	26.865999	0.621715
8	2019-02-14	27.011000	27.027000	26.865999	26.865999	-0.536822
9	2019-02-15	26.948000	26.986000	26.947001	27.027000	0.293157

Рисунок 3.12 – Розрахунок та виведення процентного змінення

Стандартне відхилення (Std Deviation). Ця ознака вимірює міру розкиду процентних змін між ціною відкриття та ціною закриття. Вона дозволяє визначити, наскільки великі відхилення від середнього можна спостерігати в рухах цін.

Більше значення стандартного відхилення може вказувати на вищий рівень невизначеності та непередбачуваності в ціновому руху.

```
mean = df['Percent Change'].mean()
number = len(df['Percent Change'])
summa = df['Percent Change'].sum()
formula = lambda x: math.sqrt((summa * (x - mean)) ** 2 / number)
df['Std Deviation'] = df['Percent Change'].apply(formula)
```

	Std Deviation
0	0.017700
1	0.032304
2	0.000938
3	0.219174
4	0.310839
5	0.029518
6	0.027228
7	0.173182
8	0.153640
9	0.080496

Рисунок 3.13 – Розрахунок та виведення стандартного відхилення

Для того щоб класифікувати волатильність на високу та низьку за допомогою процентного змінення, нам необхідно визначити поріг волатильності, наприклад, 0. Це порогове значення буде використовуватися для розділення періодів на два класи. Давайте розглянемо більш детально цей процес:

1) визначення порогового значення: поріг волатильності обирається вами відповідно до ваших цілей та особливостей даних. Наприклад, вибір значення 0 вказує на те, що будь-яке процентне змінення вважатиметься високою волатильністю;

2) створення бінарної класифікаційної ознаки: додайте новий стовпець, який буде вказувати клас волатильності. Наприклад, назвіть його 'Volatility Class'. Використовуйте умовний оператор для визначення, чи волатильність є високою чи низькою за встановленим пороговим значенням.

```
volatility_threshold = 0  # Задайте поріг волатильності
df['Volatility Class'] = (df['Percent Change'] >
volatility_threshold).astype(int)
# Виведіть оновлений датафрейм
print(df[['Date', 'Percent Change', 'Volatility Class']])
```

	Date	Percent Change	Volatility Class
0	2019-02-04	-0.054935	0
1	2019-02-05	-0.106704	0
2	2019-02-06	0.011133	1
3	2019-02-07	0.784750	1
4	2019-02-08	1.109691	1
...	...	...	...
1300	2024-01-30	0.063257	1
1301	2024-01-31	-0.110668	0
1302	2024-02-01	-0.460009	0
1303	2024-02-02	0.561425	1
1304	2024-02-03	0.111813	1

Рисунок 3.14 – Класифікація волатильності валютної пари за ключовими ознаками

Для наглядності щодо розподілу кількості класів волатильності був сформований та представлений точковий графік (рис. 3.15). Ця візуалізація допомагає легше усвідомити співвідношення між високою та низькою волатильністю, надаючи графічну інформацію про це.

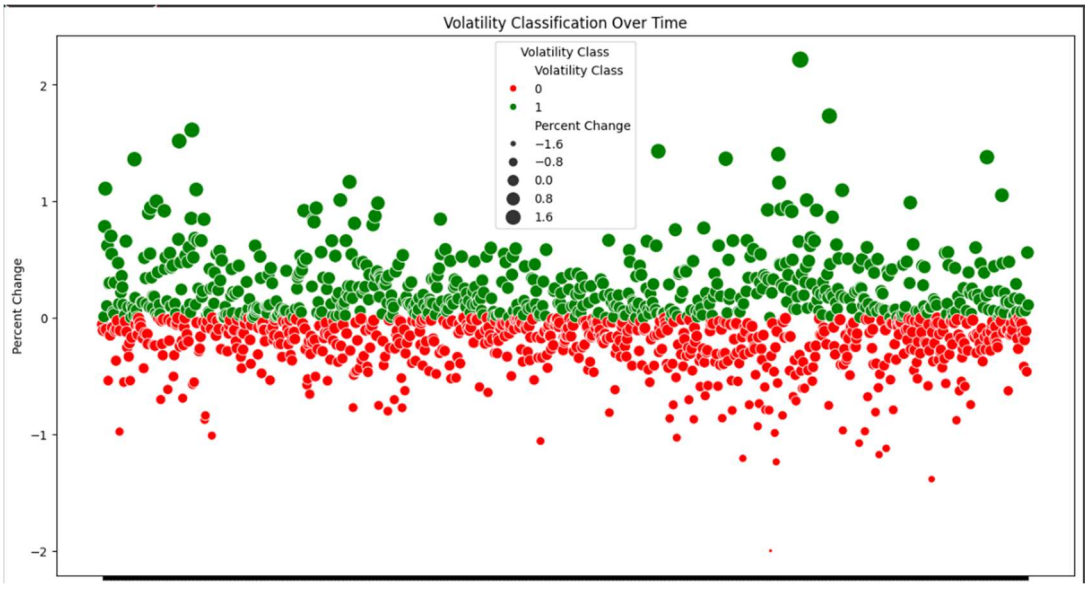


Рисунок 3.15 – Точковий графік співвідношення між класами волатильності

Наступний крок – дослідження кореляції в даних. Кореляція – це статистичний показник, який вимірює ступінь взаємозв'язку між двома змінними. У контексті дослідження даних, кореляція дозволяє визначити, наскільки сильно і в якому напрямку (позитивному або негативному) змінні змінюються разом.

Існує кілька типів кореляції, але найчастіше використовується кореляція Пірсона. Цей коефіцієнт приймає значення від -1 до 1, де:

- 1 вказує на повну позитивну лінійну залежність між змінними;
- -1 вказує на повну негативну лінійну залежність;
- 0 вказує на відсутність лінійної залежності.

Для дослідження кореляції даних у Python без графічного представлення та з виведенням результатів у консолі можна використовувати метод corr() для об'єкта DataFrame.

```
correlation_matrix = df_numeric.corr()
print("Correlation Matrix:")
print(correlation_matrix)
```

```
Correlation Matrix:
                    Open      High       Low     Close  Percent Change
Open            1.000000  0.998945  0.999799  0.999695       -0.075055
High            0.998945  1.000000  0.998860  0.998911       -0.064394
Low             0.999799  0.998860  1.000000  0.999728       -0.066099
Close           0.999695  0.998911  0.999728  1.000000       -0.050719
Percent Change -0.075055 -0.064394 -0.066099 -0.050719        1.000000
Std Deviation   0.120046  0.125776  0.115309  0.122629        0.122151

                Std Deviation
Open                 0.120046
High                 0.125776
Low                  0.115309
Close                0.122629
Percent Change       0.122151
Std Deviation        1.000000
```

Рисунок 3.16 – Матриця кореляції

Зразок кореляційної матриці, яку ви представили, має наступні значення:

1) значення кореляції між "Open", "High", "Low", та "Close" дуже високі (близькі до 1), що вказує на сильну позитивну лінійну залежність між цими змінними. Це означає, що ці ціни мають схильність рости або падати разом;

2) Percent Change – має слабку негативну кореляцію з усіма цінами ("Open", "High", "Low", "Close"). Це може вказувати на те, що існує слабка тенденція до зменшення волатильності зі збільшенням цін;

3) Std Deviation – має певний ступінь кореляції з усіма цінами, а також із "Percent Change". Це свідчить про певну залежність відхилення відсоткових змін від цін.

Тобто, ця кореляційна матриця надає вам інформацію про те, які параметри між собою корелюють, або, навпаки, не корелюють. Наприклад, висока кореляція між цінами (Open, High, Low, Close) може свідчити про те, що зміни в одному параметрі супроводжуються змінами в іншому.

Значна кореляція між Percent Change і Std Deviation може вказувати на те, що великі зміни відбуваються неоднорідно, але, скоріше за все, тут можуть виникати інші чинники, які потрібно врахувати.

Отже, встановлено, що обраний для дослідження набір даних не містить жодних нечислових, відсутніх чи аномальних значень у жодному з атрибутів. Здійснено необхідні операції заміни та нормалізації даних. Таким чином, набір даних готовий для наступного етапу – моделювання.

Моделювання – це важливий етап у процесі вирішення завдань машинного навчання, під час якого використовуються алгоритми та моделі для тренування на навчальних даних та отримання прогнозів або класифікацій на нових даних.

Другий етап підготовки до моделювання включає в себе визначення цільової змінної та визначення ознак, які будуть використовуватися для класифікації. На цьому етапі важливо розділити дані на навчальний та тестовий набори для ефективної оцінки

моделі. Дотримання балансу між класами (якщо є класифікаційна задача) та врахування стратифікації допомагають уникнути перекосу результатів.

Додатково, проводиться аналіз розподілу класів та визначення метрик якості, які будуть використовуватися для оцінки ефективності моделі. Наприкінці цього етапу дані готові до подальшого застосування алгоритмів машинного навчання та навчання моделей на навчальному наборі для подальшого тестування та оцінки їхньої точності та ефективності.

Метод відкладених даних (англ. Holdout Method) – це стратегія поділу набору даних на дві неперетинаючі частини: тренувальний (навчальний) та тестовий. Тренувальний набір використовується для навчання моделі, в той час як тестовий набір використовується для оцінки ефективності моделі та перевірки її здатності до узагальнення на нових даних.

Метод відкладених даних (рис. 3.17) дозволяє оцінити роботу моделі на реальних даних, які вона раніше не бачила, та визначити її точність та ефективність перед застосуванням до реальних сценаріїв чи вирішення завдань.

У нашому випадку, 80% даних використовується для тренування моделі, а 20% залишається для тестування. Ця процедура забезпечує наявність 1305 записів для моделювання нових прогнозів.

Рисунок 3.17 – Схематичне зображення методу відкладених даних

Модель буде навчатися на випадковій вибірці даних за допомогою функції train_test_split, де випадковим чином вибираються підмножини даних для тренування та тестування. Випадкова вибірка – це вибір певної кількості об'єктів чи прикладів з набору даних таким чином, що кожен об'єкт має однакову ймовірність бути обраним.

У контексті машинного навчання випадкова вибірка допомагає забезпечити репрезентативність даних для тренування та тестування моделі. Вона дозволяє уникнути впливу порядку даних та забезпечити, що модель отримує різноманітні приклади для навчання, що сприяє її загальній ефективності та здатності узагальнення на нові дані.

```
train_data, test_data = train_test_split(df, test_size=0.2, random_state=42,
stratify=df['Volatility Class'])
```

Якщо рандомізація виконана правильно, то розподіл класів волатильності в тренувальній та тестовій вибірках є репрезентативним та відображає вихідну кількість об'єктів кожного класу. У нашому випадку:

```
Розмір навчальної вибірки: 1044
Розмір тестової вибірки: 261
Навчальна вибірка:
Volatility Class
0    538
1    506
Name: count, dtype: int64
Тестова вибірка:
Volatility Class
0    134
1    127
Name: count, dtype: int64
```

Рисунок 3.18 – Розділення даних для тренування та тестування моделі

	Open	High	Low	Close	Percent Change	Std Deviation
839	29.283529	29.283529	29.104601	29.286367	0.009691	0.000531
255	24.152351	24.332207	24.152351	24.152788	0.001809	0.001692
493	28.170097	28.170097	28.132074	28.209843	0.141093	0.037599
334	26.356882	26.356882	26.348791	26.413706	0.215595	0.058616
190	24.956377	24.956377	24.856165	24.891697	-0.259172	0.075315
...	...	...	...	...	...	...
1111	36.858814	36.858814	36.670948	36.663483	-0.529944	0.151699
888	29.454685	29.454685	29.228907	29.221329	-0.792254	0.225697
371	26.719728	26.719728	26.643353	26.791519	0.268682	0.073592
463	27.892025	27.956528	27.892025	27.928766	0.131726	0.034957
1072	36.417641	36.502876	36.417641	36.456516	0.106748	0.027910

Рисунок 3.19 – Рандомізація даних

Тепер, коли дані підготовлені та розподілені на тренувальний та тестовий набори, можна приступати до побудови та навчання моделей наївного баєсівського класифікатора і моделі машинного навчання на основі методу опорних векторів (SVM).

3.4 Застосування простих класифікаторів для вирішення задачі класифікації

У якості базових моделей обрано моделі: наївний баєсівський класифікатор (NB) та метод опорних векторів (SVM).

Два підходи, зокрема наївний баєсівський класифікатор та SVM, використовуються для класифікації волатильності на низьку та високу. Вони навчаються на тренувальному наборі (1044 записи), який містить дані з відомими

класами, і потім випробовуються на тестовому наборі (261 запис) для оцінки їхньої ефективності та здатності узагальнення на нові дані.

На завершальному етапі проведемо оцінку ефективності обох моделей за допомогою різних метрик, таких як коефіцієнт успішності, точність, повнота, та інші. Це допоможе визначити, яка модель краще справляється з класифікацією волатильності на тестовому наборі даних.

Наївний баєсівський класифікатор (NB). Ініціалізація та навчання моделі наївного баєсівського класифікатора є важливим етапом у процесі машинного навчання. Використовуючи бібліотеку scikit-learn в Python, цей процес може бути легко виконаний за кілька кроків.

Спочатку імпортуємо необхідну бібліотеку для наївного баєсівського класифікатора:

Створюємо екземпляр класу GaussianNB(), який представляє наївний баєсівський класифікатор з гаусовим розподілом:

```
nb_classifier = GaussianNB()
```

Навчаємо модель на тренувальних даних за допомогою методу fit, передаючи тренувальні ознаки та відповідні цільові значення:

```
nb_classifier.fit(X_train, y_train)
```

де X_train – це тренувальні ознаки, а y_train – це відповідні цільові значення.

Далі ми використовуємо наївний баєсівський класифікатор (nb_classifier), який був ініціалізований та навчений на тренувальному наборі даних (X_train, y_train). Він застосовує цю модель до тестового набору даних (X_test) з метою прогнозування класу (y_pred_test). Ці прогнози базуються на вивчених закономірностях, які модель виявила під час тренування. Таким чином, y_pred_test містить прогнозовані класи для відповідних записів у тестовому наборі даних.

```
y_pred_test = nb_classifier.predict(X_test)
```

	Прогнозоване	Фактичне
1078	0	0
1119	0	1
1180	0	0
549	0	0
344	0	0
Predicted	0	1

Щоб додати порівняння результатів класифікації з даними навчання, можна використати той самий звіт про класифікацію для навчального набору даних.

```
y_pred_train = nb_classifier.predict(X_train)
```

	Прогнозоване	Фактичне
838	0	1
254	1	1
492	1	1
333	1	1
189	0	0

В бібліотеці scikit-learn, на відміну від інших бібліотек, таких як TensorFlow чи PyTorch, немає вбудованої функції summary, яка б надавала підсумкову інформацію про модель. Проте, ви можете отримати інформацію про модель, використовуючи доступні методи та атрибути.

Наприклад, для моделей, які знаходяться в бібліотеці scikit-learn, ви можете вивести параметри моделі за допомогою методу get_params(), який повертає словник з параметрами моделі та їх значеннями.

```
print("Параметри моделі:", nb_classifier.get_params())
```

Щоб з'ясувати, чи добре навчена модель, необхідно звірити її з даними навчання. Результат отримуємо як таблиці.

Predicted	0	1
Actual		
0	532	6
1	59	447

Результат показує, що помилка (або рівень помилкової класифікації) становить 65 із 1044, або 6%. Таким чином, точність класифікації на тренувальних даних – 94%.

Для відображення точності класифікації тестового набору ми сформуємо і виведемо матрицю невідповідностей (confusion matrix). Це дасть нам детальну інформацію про те, як модель класифікує дані і які помилки вона може допускати. Результати матриці невідповідностей надають можливість обчислити такі метрики, як коефіцієнт успішності (accuracy), повноту (recall), специфічність (specificity), та інші, що допомагає отримати повну картину ефективності класифікатора.

```
# Обчислити confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
# Візуалізація confusion matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

Матриця невідповідностей представлена у наступному вигляді (рис. 3.20)

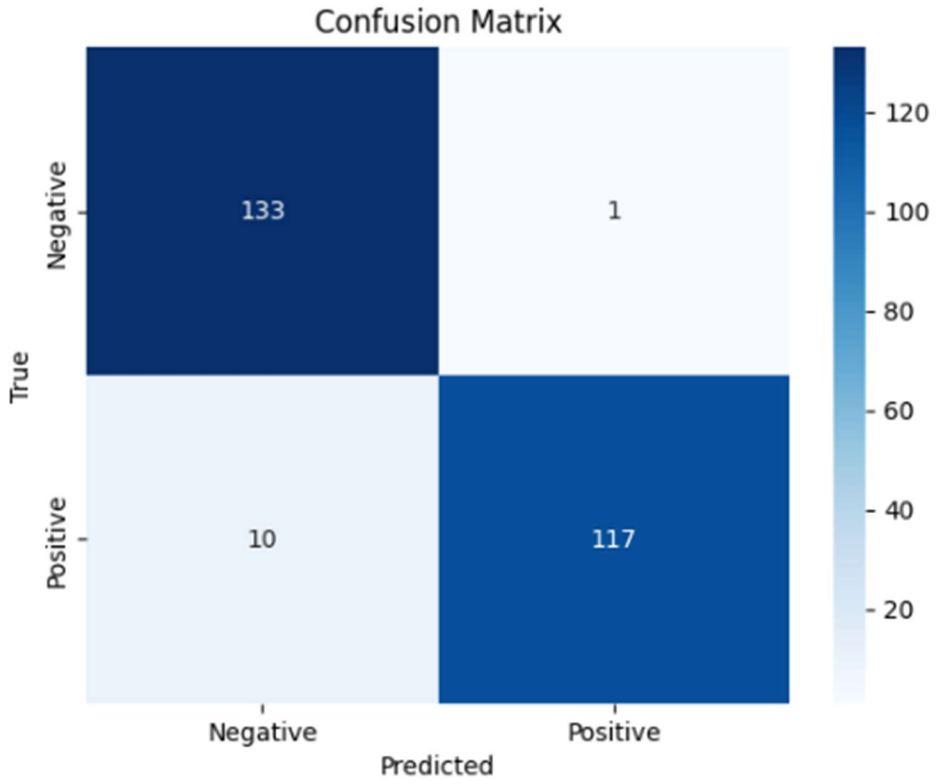


Рисунок 3.20 – Матриця невідповідностей для тестового набору даних (NB)

Розглянемо кожне значення більш детально:

- 133 (Actual Class 0, Predicted Class 0): кількість випадків, коли модель правильно визначила, що елементи належать до класу 0 (низька волатильність);
- 1 (Actual Class 0, Predicted Class 1): кількість випадків, коли модель помилилася, визначивши, що елементи належать до класу 1 (висока волатильність), тоді як насправді вони належать до класу 0;
- 10 (Actual Class 1, Predicted Class 0): кількість випадків, коли модель помилилася, визначивши, що елементи належать до класу 0, тоді як насправді вони належать до класу 1;
- 117 (Actual Class 1, Predicted Class 1): кількість випадків, коли модель правильно визначила, що елементи належать до класу 1.

Матриця невідповідностей використовується для визначення точності та частоти помилок.

Результат показує, що помилка становить 11 з 261, або 4 %. Отже, точність класифікації на тестових даних становить приблизно 96%.

Виведення звіту з класифікації для навчального та тестового набору даних спрямовані на оцінку продуктивності моделі класифікації та надає комплексну інформацію про ефективність моделі. Основні метрики включають precision, recall та f1-score для кожного класу (0 і 1), а також коефіцієнт успішності (accuracy) та середні значення цих метрик.

Звіт з класифікації для навчального набору даних:

	precision	recall	f1-score	support
0	0.90	0.99	0.94	538
1	0.99	0.88	0.93	506
accuracy			0.94	1044
macro avg	0.94	0.94	0.94	1044
weighted avg	0.94	0.94	0.94	1044

Рисунок 3.21 – Звіт з класифікації для навчального набору даних (NB)

Звіт з класифікації для навчального набору даних відображає високий рівень коефіцієнту успішності моделі, який складає 94%. Для класу 0 (низька волатильність) відзначається висока точність на рівні 90%, що свідчить про ефективність класифікації валютних пар з низькою волатильністю. Для класу 1 (висока волатильність) досягнута вражаюча точність на рівні 99%, що є дуже високим показником. Проте показник повноти (recall) для класу 1 складає лише 88%, що може вказувати на те, що деякі випадки високої волатильності можуть бути пропущені моделлю. Середній гармонічний показник F1-score для обох класів також є високими,

становлячи 0,94 для класу 0 і 0,93 для класу 1, що підтверджує ефективність моделі в обох випадках.

Звіт з класифікації для тестового набору даних:

	precision	recall	f1-score	support
0	0.93	0.99	0.96	134
1	0.99	0.92	0.96	127
accuracy			0.96	261
macro avg	0.96	0.96	0.96	261
weighted avg	0.96	0.96	0.96	261

Рисунок 3.22 – Звіт з класифікації для тестового набору даних (NB)

Звіт з класифікації для тестового набору даних показує, що коефіцієнт успішності складає 96%. Для класу 0 (низька волатильність) точність становить 93%, а для класу 1 (висока волатильність) – 99%. Показник повноти (recall) для класу 0 складає 99%, а для класу 1 – 92%. Середній гармонічний показник F1-score для класу 0 дорівнює 0,96, а для класу 1 – 0,96. Об'єднана макро-середня точність, повнота і F1-score становить 96%.

На основі наведених звітів з класифікації для навчального та тестового наборів даних можна зробити висновок, що модель демонструє досить високу ефективність у класифікації волатильності.

Коефіцієнт успішності (accuracy) для навчального набору даних складає 94%, що означає, що модель правильно класифікує близько 94% випадків. Для класу 0 точність становить 90%, а для класу 1 – 99%, що свідчить про те, що модель досить добре впоралася з обома класами, хоча точність для класу 1 є незначно вищою. Показники повноти для обох класів також високі (99% для класу 0 та 88% для класу

1), а значення середнього гармонічного показника F1-score підтверджують добру збалансованість моделі.

Для тестового набору даних коефіцієнт успішності становить 96%, що також є досить високим показником. Точність для обох класів є високою, а значення повноти та F1-score також підтверджують добру працездатність моделі на тестових даних.

Отже, можна зробити висновок, що модель є ефективною та добре узгодженою для класифікації волатильності на основі навчального та тестового наборів даних.

У фінальному відрізку, проводиться візуалізація метрик класифікації для оцінки ефективності моделі (рис. 3.23). Кожен з чотирьох стовпців на стовпчастій діаграмі представляє важливий аспект класифікації. Колірні градієнти відтінків від синього до червоного додають візуальний контраст та виділяють кожну метрику. Ця графіка створює компактний та зрозумілий зразок ефективності моделі за різними аспектами, сприяючи аналізу та порівнянню її класифікаційної здатності.

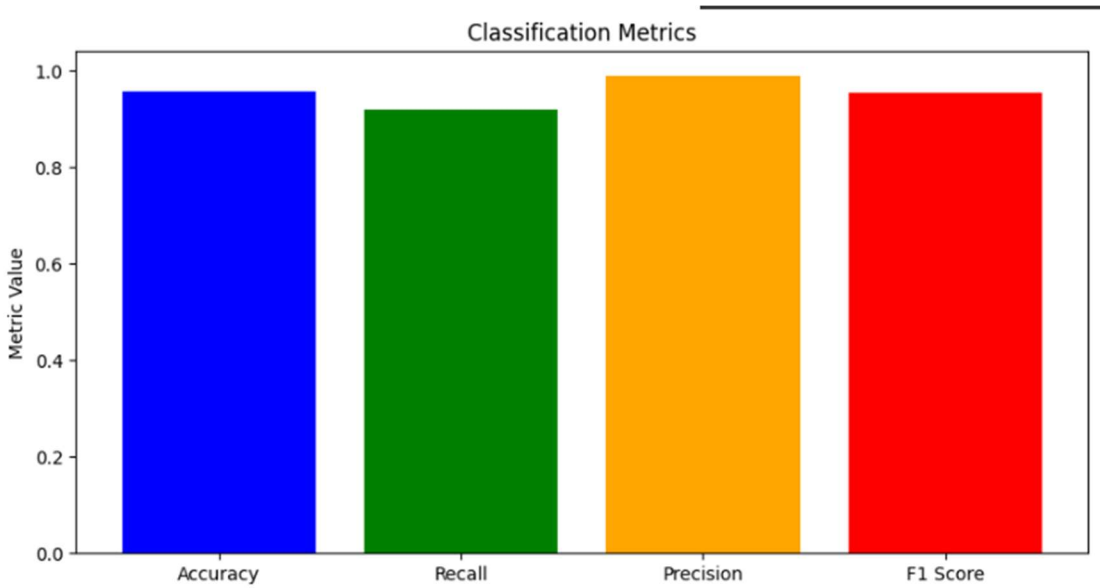


Рисунок 3.23 – Візуалізація метрик класифікації для тестового набору даних (висока волатильність)

Метод опорних векторів. Для навчання моделі методом опорних векторів (SVM) використовується клас SVC (Support Vector Classifier). Цей клас представляє собою реалізацію методу опорних векторів у бібліотеці scikit-learn.в Python.

```
from sklearn.svm import SVC
```

Метод опорних векторів (SVM) - це потужний і ефективний алгоритм машинного навчання, який широко використовується для класифікації та регресії.

Важливою складовою SVM є вибір ядра, яке визначає функцію відстані між точками в просторі ознак. Це важливий параметр, оскільки він визначає, як модель розділить класи.

У бібліотеці scikit-learn можна вибрати різні типи ядер, такі як лінійне, поліноміальне, радіальне базисне функції (RBF) тощо. Параметр ядра можна встановити за допомогою аргумента kernel при ініціалізації об'єкта SVC. Проте у бібліотеці scikit-learn за замовчуванням ядро для методу опорних векторів (SVM) - це радіальне базисне ядро (RBF), також відоме як гаусове ядро. Тому, якщо ви не вказали ядро явно, SVM буде використовувати RBF-ядро для класифікації даних.

Дії для навчання та використання моделі методу опорних векторів (SVM) подібні до дій для наївного баєсівського класифікатора.

Перед тим, як робити прогнози, модель SVM повинна бути навчена на навчальному наборі даних, щоб вона могла зрозуміти взаємозв'язки між ознаками та їх класифікаційні мітки. Цей процес включає в себе побудову оптимальної границі розділення між класами.

```
# Ініціалізуйте та навчіть модель SVM
svm_classifier = SVC()
svm_classifier.fit(X_train, y_train)
```

Коли модель SVM навчена, ми можемо використовувати її для здійснення прогнозів на нових даних. Це робиться за допомогою методу **predict**, який застосовує

навчену модель до тестового набору даних та передбачає класифікаційні мітки для кожного елементу в цьому наборі.

```
# Зробіть прогнози для тестового набору даних
y_pred_svm = svm_classifier.predict(X_test)
```

Таким чином, після успішного навчання моделі SVM ми можемо здійснювати прогнози для тестового набору даних, щоб оцінити її ефективність та точність на нових наборах даних.

	Прогнозоване	Фактичне
1078	0	0
1119	0	1
1180	0	0
549	0	0
344	0	0

Щоб додати порівняння результатів класифікації з даними навчання, можна використати той самий звіт про класифікацію для навчального набору даних.

	Прогнозоване	Фактичне
838	0	1
254	1	1
492	0	1
333	0	1
189	0	0

Для того щоб з'ясувати, чи добре навчена модель, необхідно звірити її з даними навчання. Результат отримуємо як таблиці.

Predicted	0	1
Actual		
0	491	47
1	423	83

Результат показує, що помилка (або рівень помилкової класифікації) становить 470 із 1044, або 45%. Таким чином, точність класифікації на тренувальних даних – 55%.

Для відображення точності класифікації тестового набору ми сформуємо і виведемо матрицю невідповідностей (confusion matrix). Це дасть нам детальну інформацію про те, як модель класифікує дані і які помилки вона може допускати.

Матриця невідповідностей представлена у наступному вигляді (рис. 3.24).

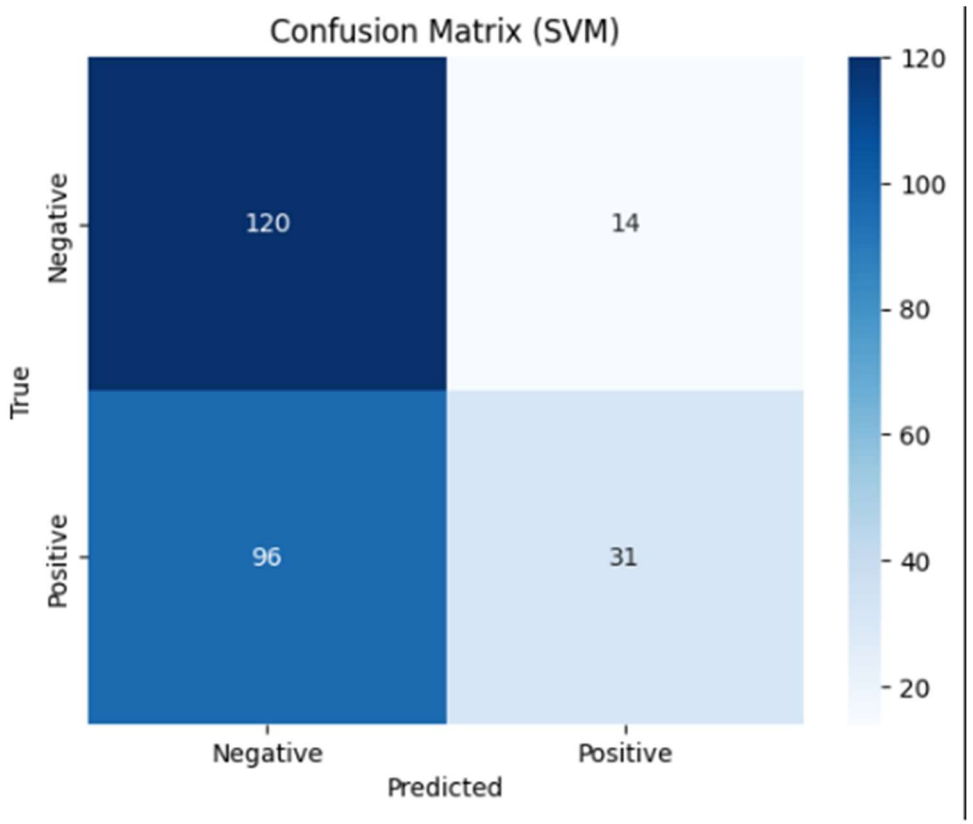


Рисунок 3.24 – Матриця невідповідностей для тестового набору даних (SVM)

Розглянемо кожне значення більш детально:

– 120 (Actual Class 0, Predicted Class 0): кількість випадків, коли модель правильно визначила, що елементи належать до класу 0 (низька волатильність);

– 14 (Actual Class 0, Predicted Class 1): кількість випадків, коли модель помилилася, визначивши, що елементи належать до класу 1 (висока волатильність), тоді як насправді вони належать до класу 0;

– 96 (Actual Class 1, Predicted Class 0): кількість випадків, коли модель помилилася, визначивши, що елементи належать до класу 0, тоді як насправді вони належать до класу 1;

– 31 (Actual Class 1, Predicted Class 1): кількість випадків, коли модель правильно визначила, що елементи належать до класу 1.

Ці значення допомагають нам зрозуміти, як модель класифікує дані та де вона може робити помилки. Наприклад, можемо побачити, що модель схильна до помилок у визначенні класу 0 (низька волатильність), плутаючи його з класом 1 (висока волатильність), але досить точно визначає клас 1.

Результат показує, що помилка становить 110 з 261, або 42 %. Отже, точність класифікації на тестових даних становить приблизно 58%.

Виведення звіту з класифікації для навчального та тестового набору даних спрямовані на оцінку продуктивності моделі класифікації та надає комплексну інформацію про ефективність моделі. Основні метрики включають precision, recall та f1-score для кожного класу (0 і 1), а також коефіцієнт успішності (accuracy) та середні значення цих метрик.

Звіт з класифікації для навчального набору даних:

	precision	recall	f1-score	support
0	0.54	0.91	0.68	538
1	0.64	0.16	0.26	506
accuracy			0.55	1044
macro avg	0.59	0.54	0.47	1044
weighted avg	0.59	0.55	0.48	1044

Рисунок 3.25 – Звіт з класифікації для навчального набору даних (SVM)

Звіт з класифікації для навчального набору даних показує, що коефіцієнт точності моделі складає 55%, що вказує на помірний рівень правильних прогнозів. Для класу 0 (низька волатильність) відзначається нижча точність на рівні 54%, що свідчить про те, що модель має тенденцію помилитися у визначенні валютних пар з низькою волатильністю. Для класу 1 (висока волатильність) точність становить 64%, що є трохи вище, але цей клас має низький показник повноти (recall) на рівні 16%, що може вказувати на те, що багато випадків високої волатильності помічено як низька. Середній гармонічний показник F1-score для обох класів також є низькими, становлячи 0,68 для класу 0 і 0,26 для класу 1, що підкреслює помірну ефективність моделі.

Звіт з класифікації для тестового набору даних:

	precision	recall	f1-score	support
0	0.56	0.90	0.69	134
1	0.69	0.24	0.36	127
accuracy			0.58	261
macro avg	0.62	0.57	0.52	261
weighted avg	0.62	0.58	0.53	261

Рисунок 3.26 – Звіт з класифікації для тестового набору даних (SVM)

Звіт з класифікації для тестового набору даних показує, що коефіцієнт успішності моделі складає 58%, що вказує на помірний рівень правильних прогнозів. Для класу 0 (низька волатильність) відзначається помірна точність на рівні 56%, що свідчить про ефективність класифікації валютних пар з низькою волатильністю. Для класу 1 (висока волатильність) точність становить 69%, що трохи вище, але цей клас має низький показник повноти (recall) на рівні 24%, що може вказувати на те, що багато випадків високої волатильності помічено як низька. Середній гармонічний показник F1-score для обох класів також є низькими, становлячи 0,69 для класу 0 і 0,36 для класу 1, що підкреслює помірну ефективність моделі.

Обидва звіти з класифікації показують, що модель має проблеми з коректною класифікацією високо-волатильних валютних пар. Хоча точність класифікації для низько-волатильних валютних пар є вищою, для високо-волатильних валютних пар вона значно нижча. Модель схильна до помилкового класифікування високо-волатильних валютних пар як низько-волатильних, що підтверджується низьким показником recall для класу 1. Загальна ефективність моделі оцінюється за допомогою середнього гармонічного показника F1-score, який для високо-волатильних валютних пар є низьким у обох випадках. Ці результати вказують на те, що модель потребує

подальшої оптимізації або використання інших методів для покращення її здатності до класифікації високо-волатильних валютних пар.

У фінальному відрізку, проводиться візуалізація метрик класифікації для оцінки ефективності моделі (рис. 3.27). Кожен з чотирьох стовпців на стовпчастій діаграмі представляє важливий аспект класифікації. Колірні градієнти відтінків від синього до червоного додають візуальний контраст та виділяють кожну метрику. Ця графіка створює компактний та зрозумілий зразок ефективності моделі за різними аспектами, сприяючи аналізу та порівнянню її класифікаційної здатності.

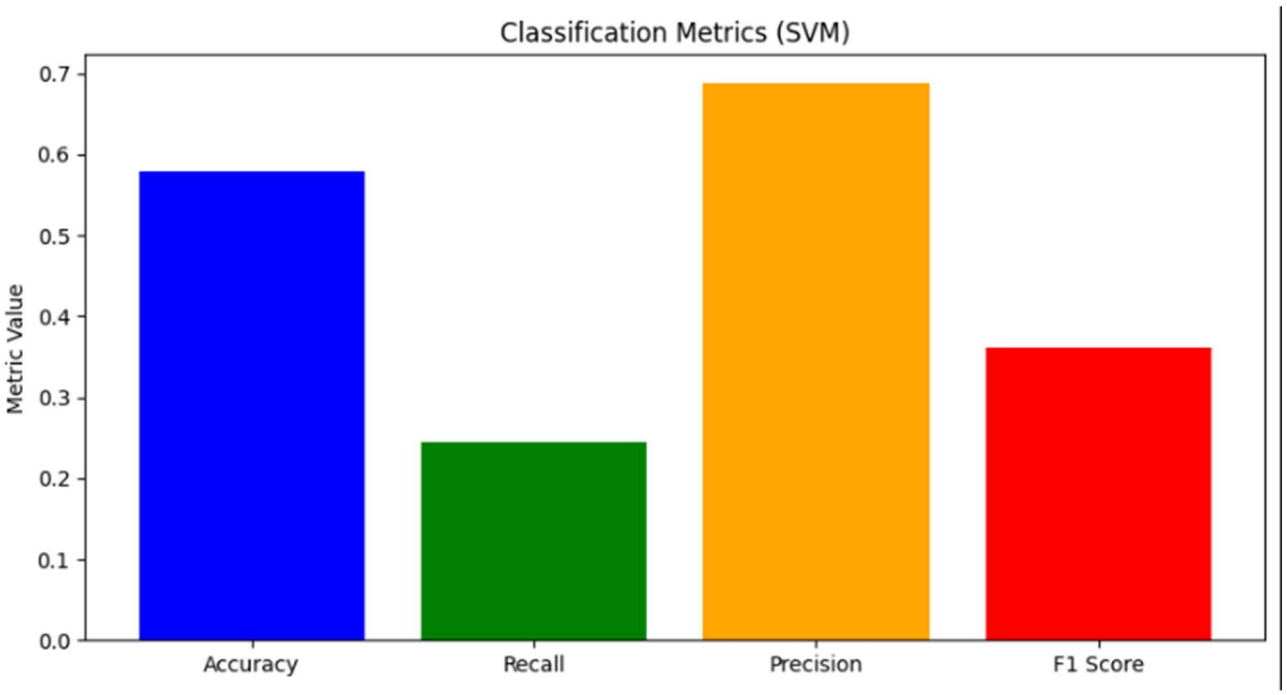


Рисунок 3.27 – Візуалізація метрик класифікації для тестового набору даних (висока волатильність)

3.5 Порівняльний аналіз роботи моделей класифікаторів

Для прийняття рішення про кращу модель для розв’язання задачі класифікації волатильності валют порівнюються результати моделювання, зібрані у таблицю 3.1.

Для оцінки ефективності кожної моделі порівнюються метрики precision, recall, F1-score для обох класів волатильності (низька та висока) та accuracy.

Accuracy (коефіцієнт успішності) – це метрика, що оцінює загальну ефективність моделі без розгляду на розподіл класів. Вона визначає відсоток правильно класифікованих екземплярів серед усіх екземплярів у тестовому наборі даних.

Чим вищі значення цих метрик, тим краще модель класифікує валютні пари. Нижче наведена порівняльна таблиця з результатами моделювання.

Таблиця 3.1 – Порівняльна таблиця методів класифікації

Тип моделей	Коефіцієнт успішності (accuracy)	Точність (precision) клас 0	Точність (precision) клас 1	Повнота (recall) клас 0	Повнота (recall) клас 1	F-міра (F-measure) клас 0	F-міра (F-measure) клас 1
Наївний Баєсів	0.96	0.93	0.99	0.99	0.92	0.96	0.96
Метод опорних векторів	0.58	0.56	0.69	0.90	0.24	0.69	0.36

За результатами оцінки ефективності моделей для задачі класифікації волатильності валют, найкращою моделлю виявився наївний баєсівський класифікатор. Це відображено вищеуказаними метриками ефективності. Наївний

Баєсів класифікатор демонструє кращі значення коефіцієнта успішності (accuracy), точності (precision), повноти (recall), а також F-міри (F-measure) порівняно з методом опорних векторів. Таким чином, на основі сукупності цих показників обрано наївний баєсівський класифікатор як кращу модель для розв'язання задачі класифікації волатильності валют.

Декілька факторів можуть впливати на ефективність моделі методу опорних векторів (SVM):

1) параметри ядра: вибір підходящого ядра (наприклад, лінійного, поліноміального або RBF) може вплинути на результати моделі. Для деяких наборів даних одне ядро може працювати краще за інше. Уточнення параметрів ядра, таких як степінь полінома або гамма в RBF ядрі, може поліпшити ефективність моделі;

2) наявність шуму в даних: наявність шуму в навчальних даних може призвести до перенавчання моделі, що може погіршити її ефективність на тестових даних. Одним із способів управління шумом є використання технік очищення даних або регуляризації моделі;

3) наявність викидів: наявність викидів або аномалій у навчальних даних також може вплинути на ефективність моделі. Використання методів виявлення та видалення викидів може покращити результати моделі;

4) розмір навчального набору: модель SVM може вимагати значного обсягу навчальних даних для навчання ефективної границі рішень. У випадку обмеженого обсягу даних може бути корисним застосування методів зменшення розмірності даних або аугментації даних для збільшення навчального набору;

5) неузгодженість у виборі гіперпараметрів: неправильний вибір гіперпараметрів, таких як параметри регуляризації або штрафи за помилки класифікації, може призвести до невідповідності моделі даним. Використання методів пошуку гіперпараметрів, таких як перехресна перевірка, може допомогти знаходити оптимальні значення цих параметрів.

Усунення цих проблем може включати в себе експерименти з різними конфігураціями моделі, використання більш складних методів очищення та підготовки даних, а також уточнення параметрів моделі через методи оптимізації гіперпараметрів.

3.6 Реалізація обгортки інтелектуальної системи – Телеграм-боту

Інтелектуальна система класифікації волатильності валют, реалізована у вигляді телеграм-бота, представляє собою комплексний інструмент для аналізу та класифікації волатильності на ринку валют. Цей бот був розроблений використовуючи набір методів та технологій, які описані раніше у даній роботі.

В першу чергу, для побудови системи була проведена обробка вхідних даних. Ці дані були попередньо очищені, відфільтровані та підготовлені для подальшого використання. Під час обробки даних також було проведено ряд додаткових етапів, таких як нормалізація, розбиття на навчальний та тестовий набори, а також визначення цільової змінної для класифікації.

Після обробки даних ми приступили до моделювання. Для класифікації волатильності була обрана модель наївного баєсового класифікатора, яка показала добрі результати у попередніх дослідженнях. Модель була навчена на навчальному наборі даних та перевірена на тестовому наборі для оцінки її ефективності та прогностичної здатності.

Нарешті, після побудови та валідації моделі, інтелектуальна система була успішно впроваджена в середовище телеграм-боту з метою надання користувачам зручного та ефективного інструменту для відстеження та аналізу валютних ринків. Використання телеграм-боту дозволяє забезпечити простий та доступний інтерфейс для користувачів, які можуть отримувати актуальну інформацію та аналіз валютних курсів у режимі реального часу.

Основні переваги впровадження інтелектуальної системи в телеграм-бот включають:

1) зручний доступ: користувачі можуть отримувати інформацію про волатильність валют та їх класифікацію просто взаємодіючи з телеграм-ботом. Це робить процес отримання аналітичних даних більш зручним та ефективним;

2) регулярні оновлення: система автоматично оновлює та надсилає інформацію про волатильність валютних пар, що дозволяє користувачам залишатися в курсі останніх подій на ринку;

3) інтерактивні функції: телеграм-бот може взаємодіяти з користувачами, надаючи їм можливість отримати додаткову інформацію та взаємодіяти з функціоналом системи;

4) персоналізований досвід: користувачі можуть налаштовувати отримання інформації відповідно до своїх індивідуальних потреб та вибирати конкретні валютні пари для відстеження.

Це робить високотехнологічну систему класифікації волатильності валют більш доступною та корисною для широкого кола користувачів через інтерфейс телеграм-боту.

Реєстрація телеграм-бота є основою у процесі розробки обгортки для інтелектуальної системи класифікації волатильності валют. Цей процес полягає у створенні та налаштуванні бота за допомогою спеціальних засобів, наданих платформою Telegram.

Перший крок – це створення нового бота в Telegram через спеціального бота-конструктора @BotFather. BotFather – це офіційний бот Telegram для створення та налаштування інших ботів. Користувач відправляє команду "/newbot" та слідує інструкціям для створення нового бота, вибору його імені та отримання токену доступу (рис. 3.28).

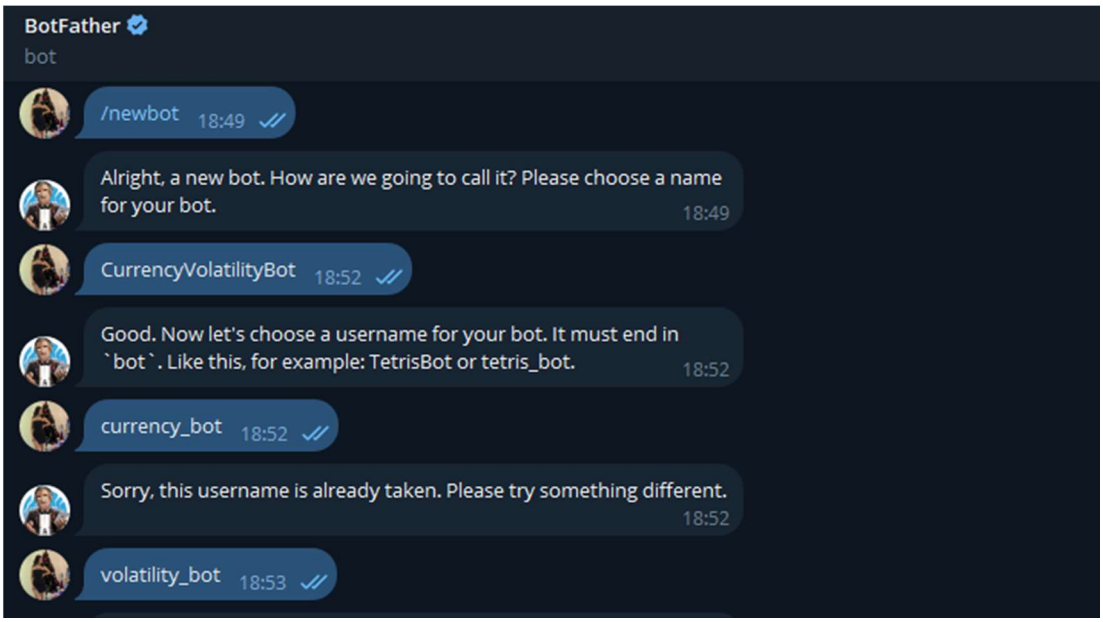


Рисунок 3.28 – Створення нового бота через команду /newbot

Команда "/mybots" використовується для перегляду списку ботів, які ви створили або адмініструєте в Telegram. Ця команда допомагає користувачам швидко отримати доступ до управління своїми ботами та їх налаштуваннями. На рисунку 3.29 відображено процес завантаження фото профілю для бота.

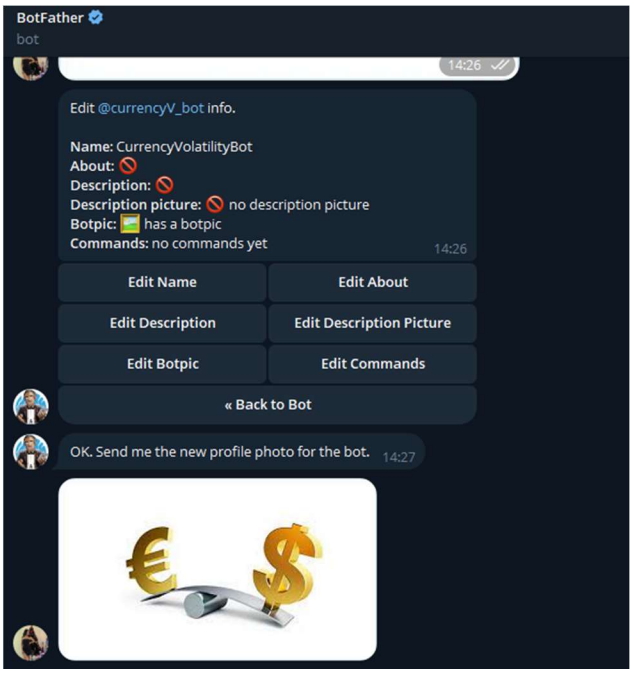


Рисунок 3.29 – Управління та налаштування бота через команду /mybots

Для взаємодії з Telegram API, розробнику необхідно отримати API-ключі. Ці ключі дозволять боту взаємодіяти з платформою Telegram, приймати та відправляти повідомлення, обробляти запити користувачів та інше.

Роль BotFather у даній ситуації полягає в виділенні унікального токену, який ідентифікує створеного бота та надає йому доступ до Telegram API. Цей токен необхідно додати до ключового файлу confyg.py (файл для зберігання конфігураційних параметрів, таких як токен бота).

Завдяки цьому процесу реєстрації, бот стає доступним для користувачів у месенджері Telegram і готовий до виконання функцій, які передбачені для системи класифікації волатильності валют.

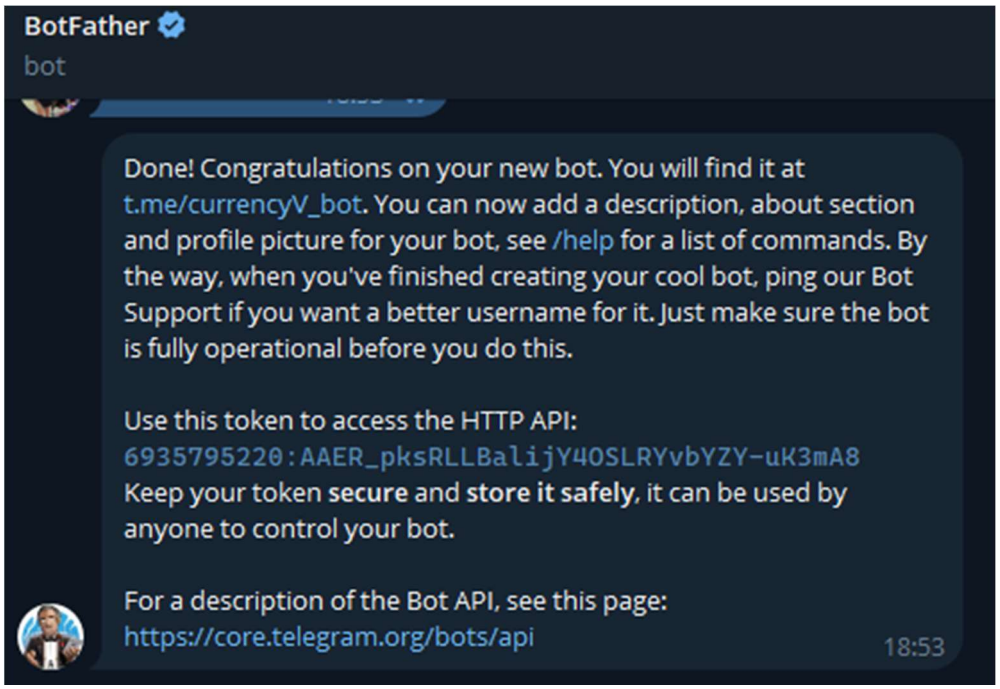


Рисунок 3.30 – Отримання API-ключа

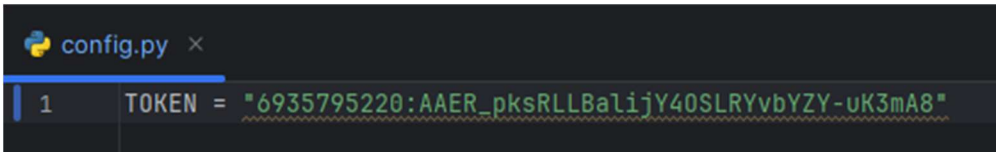


Рисунок 3.31 – Додання токена до ключового файлу

Структура проекту телеграм-боту – це організація файлів та коду, що визначає, як різні частини функціональності боту організовані та взаємодіють між собою. Нижче наведено загальну структуру проекту (рис. 3.32):

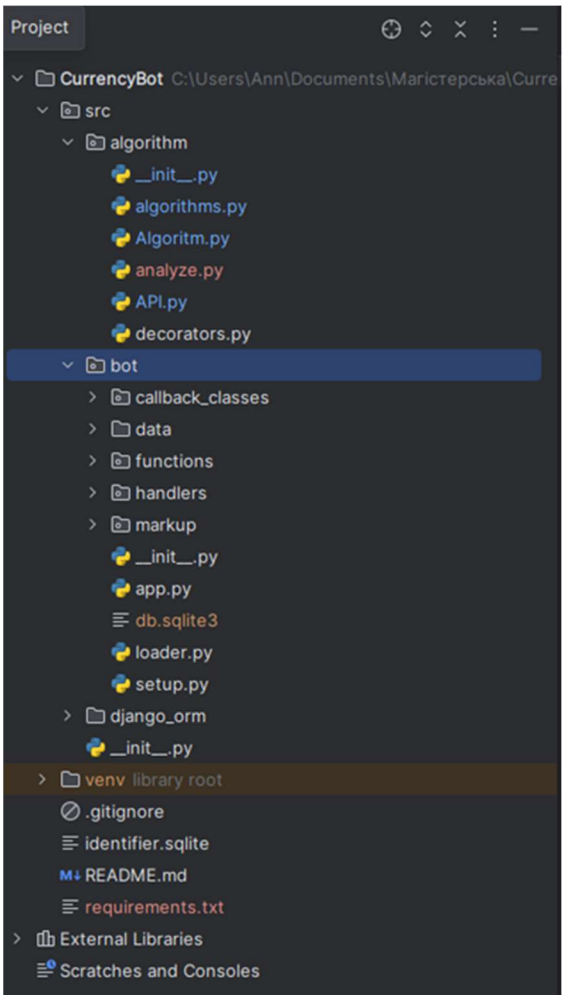


Рисунок 3.32 – Структура проекту телеграм-боту

Структура проекту телеграм-боту може включати в себе різні компоненти, які спільно забезпечують його функціональність та ефективність:

1) обробники (Handlers);
2) модулі функціональності;
3) модулі комунікації;
4) утиліти та допоміжні функції;
5) модуль бази даних;
6) модуль авторизації та безпеки;
7) модуль взаємодії з іншими сервісами.

Такий розподіл компонентів допомагає зробити структуру телеграм-боту зручною для розширення та підтримки різноманітних функцій, сприяючи взаємодії з користувачами та забезпечуючи стабільну та ефективну роботу системи.

Для початку роботи з чат-ботом необхідно надіслати команду "start". Ця команда ініціює взаємодію з ботом і дозволяє користувачеві отримати доступ до основного функціоналу.

Основний функціонал телеграм-бота включає наступні запити, які формують основу нашої системи:

1) поняття волатильності: запит дозволяє користувачам отримати роз'яснення щодо поняття волатильності на ринку валют. Це може включати пояснення, що таке волатильність, як вона вимірюється, і чому вона важлива для трейдерів та інвесторів;

2) короткий FAQ по користуванню системою: запит надає короткий перелік питань і відповідей щодо використання системи;

3) класифікація волатильності валют: запит дозволяє користувачам отримати класифікацію рівнів волатильності різних валютних пар (заздалегідь вибравши необхідні параметри роботи системи). Включає інформацію про стабільні та волатильні валютні пари, їх характеристики та фактори, що впливають на волатильність;

4) відстеження графіків валютних пар в реальному часі: запит дозволяє користувачам переглядати графіки валютних пар в реальному часі. Реалізація проводилась через інтеграцію з фінансовими API для отримання актуальних даних про ціни валют та їх зміни у часі.

Ці запити становлять основу функціоналу чат-бота і дозволяють користувачам отримувати широкий спектр інформації щодо волатильності валют та відстежувати її динаміку у реальному часі. Тож розглянемо їх більш детальніше (рис. 3.33).

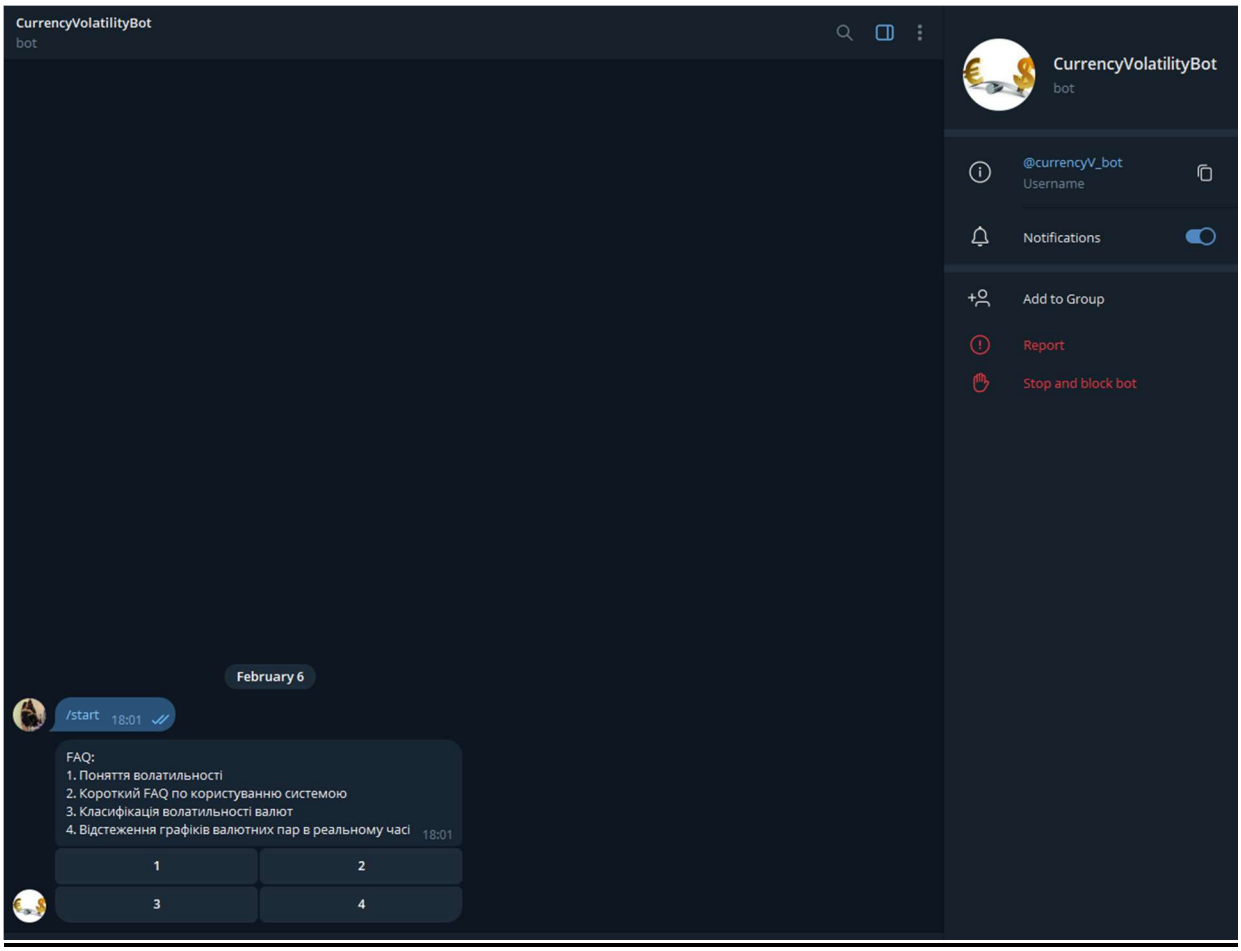


Рисунок 3.33 – Перелік запитів телеграм-боту

При виборі запиту "Поняття волатильності" користувач отримує наступну інформацію (рис. 3.34).

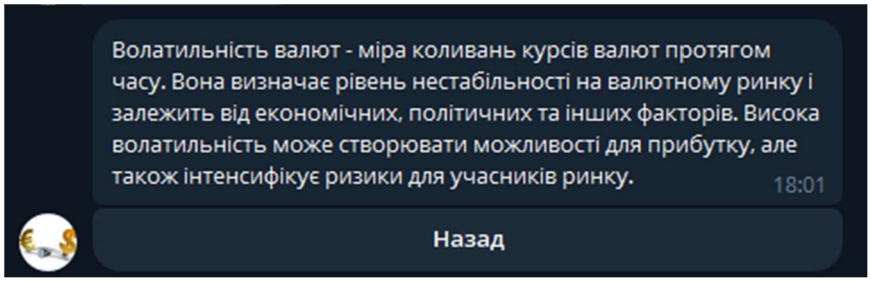


Рисунок 3.34 – Відповідь на запит «Поняття волатильності»

При виборі запиту "Класифікація волатильності валют" користувач отримує доступ до меню з наступними опціями (рис. 3.35).

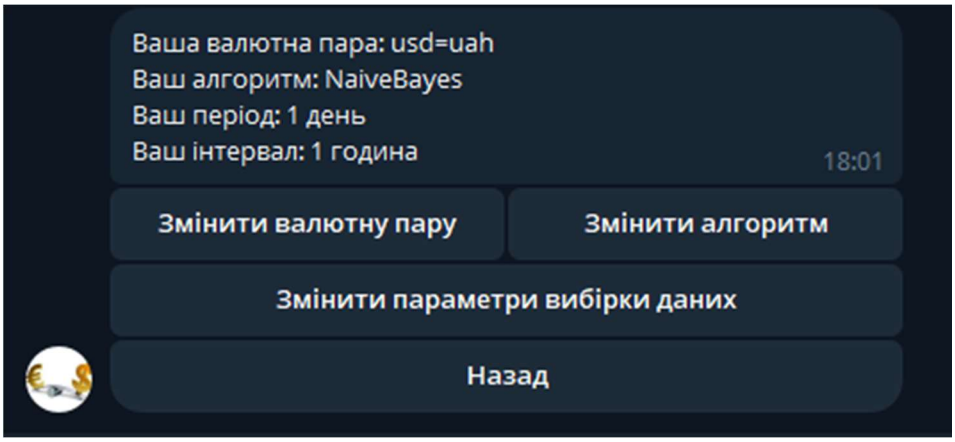


Рисунок 3.35 – Відповідь на запит «Класифікація волатильності валют»

1. «Змінити валютну пару». Користувач може обрати конкретну валютну пару, для якої буде проводитися класифікація волатильності (рис. 3.36). Наприклад, EUR/PLN, USD/UAH, USD/EUR або будь-яка інша пара валют.

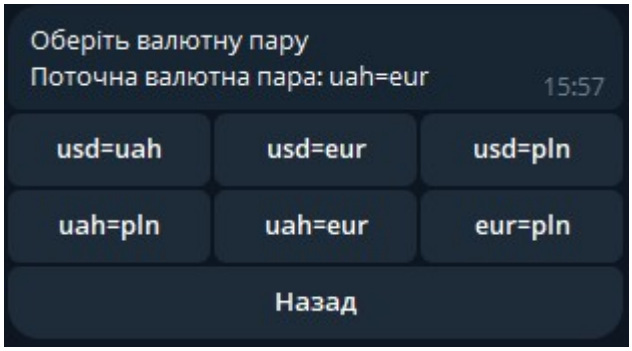


Рисунок 3.36 – Відповідь на запит «Змінити валютну пару»

2. «Змінити алгоритм». Користувач має можливість вибрати алгоритм, який буде використовуватися для класифікації волатильності (рис. 3.37). Це може бути наївний баєсівський класифікатор (NaiveBayes) або метод опорних векторів (SVM).

Якщо ми захочемо, то завжди можемо розширити можливості та варіанти вибору моделей для класифікації волатильності валют.

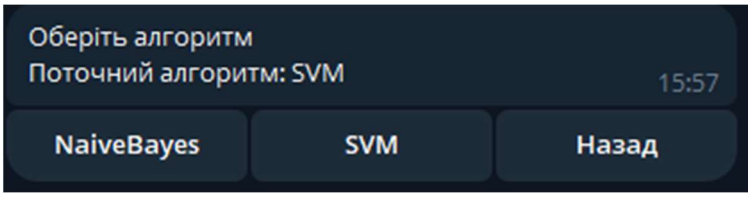


Рисунок 3.37 – Відповідь на запит «Змінити алгоритм»

3. «Змінити параметри вибірки даних». Перед нами відкривається меню, де користувач може переглянути обрані параметри та змінити період або інтервал вибірки даних (рис. 3.38). Це меню надає користувачеві можливість налаштувати параметри класифікації волатильності валют відповідно до їхніх потреб та вимог.

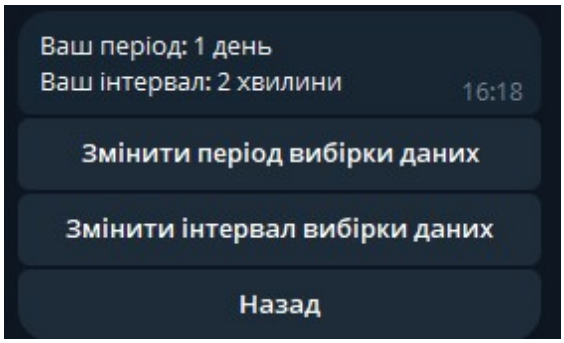


Рисунок 3.38 – Відповідь на запит «Зміиити параметри вибірки даних»

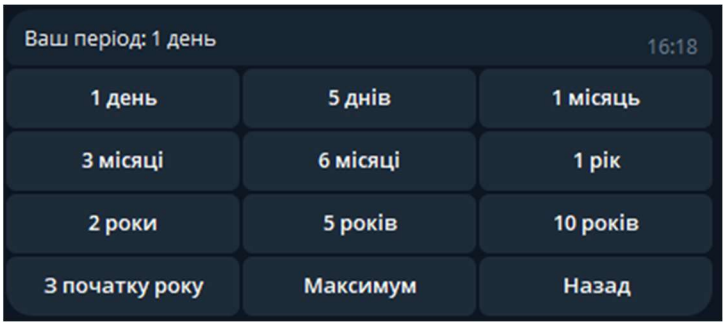


Рисунок 3.39 – Відповідь на запит «Зміиити період вибірки даних»

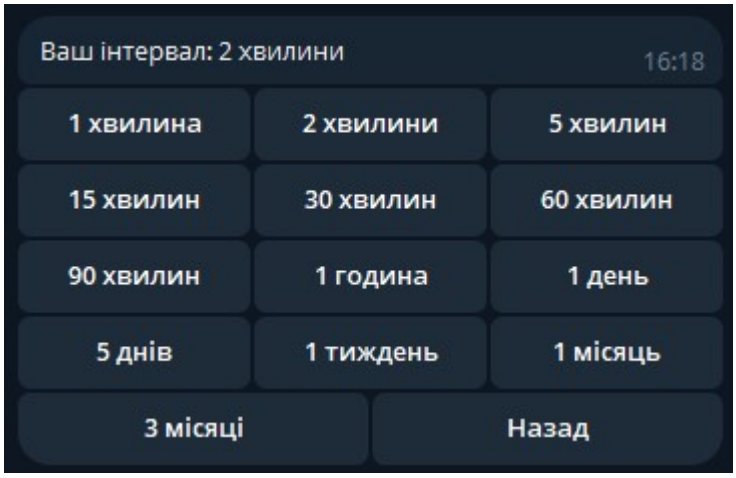


Рисунок 3.40 – Відповідь на запит «Зміиити інтервал вибірки даних»

Після проходження усіх кроків користувач має можливість натиснути кнопку "Старт", що відкриє доступ до інтерактивної таблиці з необхідними стовпцями для відстеження зростання або зниження волатильності обраної валютної пари. На основі наданої таблиці, можна розрахувати метрики для оцінки точності класифікації волатильності валютної пари.

Відтепер користувач може зручно відстежувати динаміку волатильності валютних пар та отримувати об'єктивну оцінку ефективності використовуваної моделі класифікації. Цей інструментарій дозволяє здійснювати оперативний аналіз ринкових умов та приймати обгрунтовані рішення з максимальною ефективністю.

На рисунку 3.41 представлена таблиця з даними за період 24 січня з інтервалом оновлення кожні 2 хвилини, яку використовує наївний баєсівський класифікатор для прогнозування волатильності валютних пар. Таблиця містить різні стовпці, які дозволяють відслідковувати зміни волатильності на ринку. Кожен рядок таблиці представляє собою окремий часовий відрізок, під час якого здійснювалися виміри волатильності для певних валютних пар.

Ця таблиця дозволяє нам аналізувати та візуалізувати динаміку волатильності на ринку, а також оцінювати ефективність прогнозування, здійсненого наївним баєсовим класифікатором. Шляхом вивчення цих даних можна зробити висновки щодо того, наскільки ефективно модель прогнозує волатильність та якість її прогнозів.

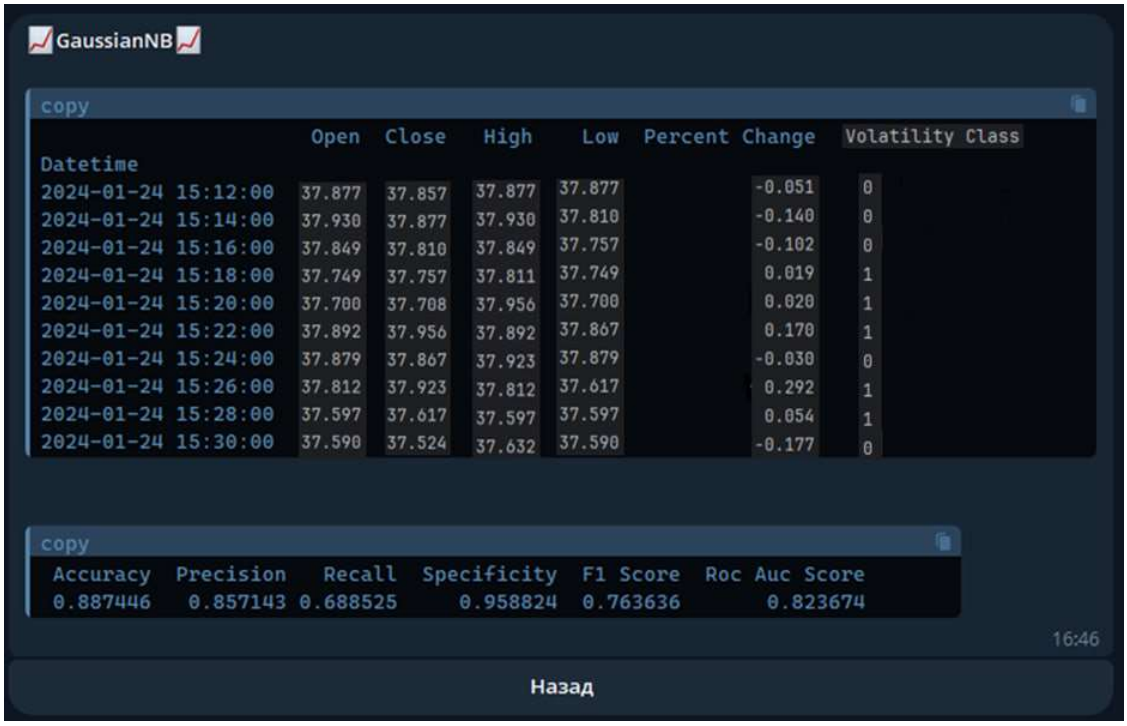


Рисунок 3.41 – Інтерактивна таблиця класифікації волатильності валютної пари

При виборі останнього запиту "Відстеження графіків валютних пар в реальному часі" ви маєте змогу спостерігати графіки, які відображають динаміку зміни вартості обраних валютних пар протягом певного періоду, що дозволяє вам більш детально вивчати їхню поведінку та прогнозувати майбутні рухи.

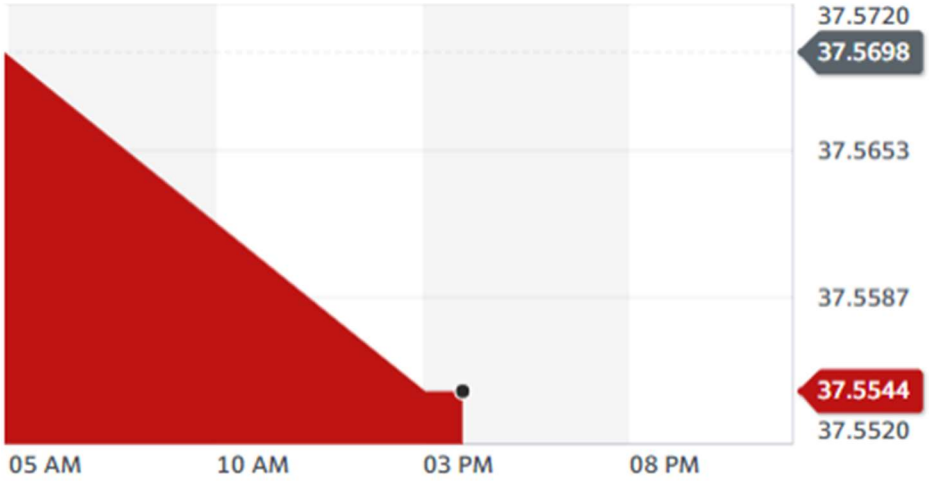


Рисунок 3.42 – Відповідь на запит «Відстеження графіків валютних пар в реальному часі»

Важливо відзначити, що графіки будуть відображати динаміку вартості саме тих валютних пар, які ви вибрали для класифікації. Це дозволяє вам не лише відстежувати зміни на ринку у реальному часі, але й порівнювати їх з результатами класифікації, щоб з'ясувати, наскільки ефективно ваша модель прогнозує рухи цих валютних пар. Такий підхід надає вам цінну інформацію для прийняття рішень на основі аналізу ринку та результатів прогнозування.

За допомогою цієї системи користувачі можуть ефективно аналізувати волатильність валютних пар, відстежувати їхню динаміку та здійснювати точні прогнози майбутніх рухів цін. Крім того, система дозволяє користувачам швидко та зручно змінювати параметри аналізу, вибирати різні алгоритми класифікації та переглядати реальний часовий графік зміни курсу валют.

Такий інструмент допомагає трейдерам та інвесторам приймати обґрунтовані рішення та досягати більшої успішності на фінансових ринках.

Висновки до розділу 3

У даному розділі було проведено детальний аналіз та розробка інтелектуальної системи для класифікації волатильності валют. Засоби розробки були оглянуті, набір даних було описано, а також проведено попередню обробку даних для підготовки до моделювання. Для вирішення задачі класифікації були застосовані прості класифікатори, такі як наївний баєсів класифікатор та метод опорних векторів (SVM), і проведений порівняльний аналіз їх роботи.

Особлива увага була приділена реалізації обгортки інтелектуальної системи у вигляді телеграм-боту, що дозволить користувачам отримувати доступ до класифікації волатильності валютних пар у зручному форматі та в реальному часі.

Загальною метою розділу було побудування ефективної системи класифікації волатильності валют з використанням сучасних методів машинного навчання та забезпечення доступу до неї через інтерфейс телеграм-боту. Результати аналізу та розробки підтвердили можливість успішної реалізації цієї системи та її корисність для користувачів, що зацікавлені в аналізі валютних ринків.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було застосовано методи інтелектуального аналізу даних та інтелектуального прийняття рішень на основі навчальної вибірки. У процесі розробки системи використовувалися передові методи машинного навчання, зокрема наївний баєсівський класифікатор та метод опорних векторів. Ці методи були застосовані для ефективної класифікації валютних пар на основі їх волатильності, що дозволяє системі робити обґрунтовані прогнози та приймати рішення на основі аналізу вхідних даних. Інтелектуальна система класифікації волатильності валют є потужним інструментом аналізу фінансових ринків, який забезпечує можливість класифікації та прогнозування коливань валютних курсів.

Ключовим етапом в розробці системи було попереднє оброблення даних, яке включало в себе очищення, нормалізацію та інші методи підготовки даних для моделювання. Далі проводився аналіз та порівняльне дослідження різних моделей класифікаторів для вибору найефективнішої.

Після реалізації моделей і створення телеграм-боту як обгортки для зручного доступу до результатів, система була готова до використання. Її користувачі можуть отримувати об'єктивні та точні прогнози щодо руху валютних ринків, що допомагає приймати обґрунтовані рішення щодо управління фінансовими портфелями та ризиками.

Метою кваліфікаційної роботи магістра була розробка та дослідження інтелектуальної системи, яка здатна ефективно класифікувати рівень волатильності на валютних ринках. Виходячи з мети, виконано поставлені завдання:

1) проведено збір та підготовку даних;

2) вивчено та досліджено методи машинного навчання та метрики якості оцінки класифікації;

3) обрано методи класифікації та побудовано базові моделі;

4) проаналізовано ефективність реалізованих методів класифікації з використанням метрик;

5) реалізовано обгортку інтелектуальної системи у вигляді телеграм-боту для взаємодії з користувачами та обробки їхніх запитів;

6) інтегровано розроблені моделі в телеграм-бот для автоматичного аналізу запитів користувачів та надання прогнозів;

7) проведено тестування розробленого телеграм-боту для перевірки його працездатності та надійності.

Таким чином, інтелектуальна система класифікації волатильності валют має змогу стати невід'ємною частиною аналітики фінансових ринків, допомагаючи її користувачам отримувати важливу інформацію та робити обдумані рішення в умовах постійної динаміки на ринках.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Зайченко Ю. П. Основи проектування інтелектуальних систем. Київ: Слово, 2006. 352 с.

2. Береза А.М. Основи створення інформаційних систем: навч. посібник. Київ: КНЕУ, 2001. 205 с.

3. Любарський С.В., Шаціло П.В. Методологія вибору моделі подання знань в інтелектуальних навчальних системах: збірник наукових праць ВІТІ НТУУ «КПІ». Київ, 2010. С. 65–71.

4. Карпенко, А. В. (2022). Телеграм як платформа для розробки чат-ботів. Вісник НГУЕУ, 1(70), 12-25 с.

5. Карпенко, А. В. (2022). Розробка чат-ботів на платформі Telegram. Матеріали міжнародної науково-практичної конференції "Інновації в інформаційних технологіях" (м. Київ, 2022 р.). Київ: Видавництво НУБіП України.

6. Telegram FAQ: веб-сайт. URL: https://telegram.org/faq (дата звернення: 27.11.2023).

7. Вимоги до оформлення дисертації : затв. наказом МОН України від 12.01.2017 No 40. URL: https://zakon.rada.gov.ua/laws/show/z0155-17 (дата звернення: 25.11.2023).

8. ТОП-40 популярних телеграм-ботів в Україні: фінанси, шопінг і відпочинок: веб-сайт. URL: https://psm7.com/uk/news/top-40-boty-telegramv-ukraine.html (дата звернення: 28.11.2023).

9. Chatbots With Machine Learning: Building Neural Conversational Agent. URL: https://dzone.com/articles/chatbotswith-machine-learning-building-neural-con (дата звернення: 30.11.2023).

10. George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung (1970) Time Series Analysis: Forecasting and Control p. 269-270.

11. Зайченко Ю. П. Основи проектування інтелектуальних систем. Київ: Слово, 2006. 352 с.

12. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean (2013) Distributed Representations of Words and Phrases and their Compositionality p. 153

13. Yahoo Finance API – A Complete Guide. URL: https://algotrading101.com/learn/yahoo-finance-api-guide/ (дата звернення: 19.12.2023).

14. Yahoo! Finance. URL: https://finance.yahoo.com (дата звернення: 19.12.2023).

15. Volatility: Meaning In Finance and How it Works with Stocks. URL: https://www.investopedia.com/terms/v/volatility.asp/ (дата звернення:03.01.2024).

16. Market and historical volatility. URL: https://wholesale.banking.societegenerale.com/en/news-insights/glossary/market-and-historical-volatility/ (дата звернення: 03.01.2024).

17. Volatility – Meaning, Types, Factors Affecting It, Calculation And Examples. URL: https://www.tickertape.in/blog/volatility/ (дата звернення: 03.01.2024).

18. Financial Market Indicators. URL: https://www.fe.training/free-resources/portfolio-management/financial-market-indicators/ (дата звернення: 04.01.2024).

19. Classification in Machine Learning: What it is & Classification Models. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning (дата звернення: 07.01.2024).

20. Types of Classification in Mschine Learning. URL: https://www.knowledgehut.com/blog/data-science/types-of-classification-in-ml (дата звернення: 07.01.2024).

21. Forecasting with Machine Learning. URL: https://www.kaggle.com/code/ryanholbrook/forecasting-with-machine-learning (дата звернення: 08.01.2024).

22. Types of Machine Learningю. URL: https://www.geeksforgeeks.org/types-of-machine-learning/ (дата звернення: 08.01.2024).

23. Binary Classification. URL: https://h2o.ai/wiki/binary-classification/ (дата звернення: 09.01.2024).

24. 4 Most Popular Machine Learning Classification Algorithms. URL: https://data-science-ua.com/blog/4-most-popular-machine-learning-classification-algorithms/ (дата звернення: 09.01.2024).

25. Гожий О.П., Калініна І.О. Методичні рекомендації до виконання курсової роботи з дисципліни «Методи та системи машинного навчання». Миколаїв: ЧНУ ім. П. Могили, 2021. 17 с.

26. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. California: Springer–Verlag, 2009. 746 p.

27. Дубініна С. В. Байєсівські методи моделювання актуальних процесів та оцінювання ризиків страхових компаній: дис. канд. техн. наук: 05.13.23. Київ, 2017. 199 с.

28. Naive Bayes Classifier. URL: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c (дата звернення: 11.01.2024).

29. Naïve Bayes. URL: https://scikit-learn.org/stable/modules/naive_bayes.html (дата звернення: 11.01.2024).

30. Ethem Alpaydin. Introduction To Machine Learning. 3th ed. Massachusetts, 2009. 584 p.

31. Support Vector Machines. URL: https://scikit-learn.org/stable/modules/svm.html (дата звернення: 13.01.2024).

32. 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: 13.01.2024).

33. Шеремет О.І., Садовой О. В. Метод опорних векторів (SVM): наук. журнал “Математичні моделі“. Донецьк, 2013. С. 13–17.

34. Бідюк П.І., Кузнєцова Н.В., Терентьєв О.М. Система підтримки прийняття рішень для аналізу даних. Київ: Наукові вісті НТУУ «КПІ», 2011. С. 48– 61.

35. Метрики в задачах машинного навчання. URL: https://habr.com/ru/company/ods/blog/328372 (дата звернення: 15.01.2024).

36. Ставицький А.В. Класифікаційні метрики. URL: http://www.andriystav.cc.ua/Downloads/MITER/Lecture_04.pdf (дата звернення: 15.01.2024).

37. Libraries for Python. URL: https://www.geeksforgeeks.org/libraries-in-python/ (дата звернення: 20.01.2024).

38. An introduction to machine learning with scikit-learn. URL: https://scikit-learn.org/stable/tutorial/basic/tutorial.html (дата звернення: 27.01.2024).

39. Preparing Your Dataset for Machine Learning: 10 Basic Techniques That Make Your Data Better. URL: https://www.altexsoft.com/blog/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better/ (дата звернення: 02.02.2024).

40. Hazewinkel, Michiel, ред. (2001). Correlation (in statistics). Енциклопедія математики[en] . Springer. ISBN.

41. Borsdorf, Rudiger; Higham, Nicholas J.; Raydan, Marcos (2010). Computing a Nearest Correlation Matrix with Factor Structure.. SIAM J. Matrix Anal. Appl. 31 (5): 2603–2622.

ДОДАТОК А

Програмна реалізація інтелектуальної системи

algorithms.py

```
from dataclasses import dataclass

from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC


@dataclass
class AlgorithmsList:
    NaiveBayes = GaussianNB
    SVM = SVC


class AlgorithmsDict:
    def __init__(self):
        self.algorithms = {
            "NaiveBayes": AlgorithmsList.NaiveBayes,
            "SVM": AlgorithmsList.SVM
        }

    def get(self, algorithm: str):
        return self.algorithms.get(algorithm)
```

algorithms.py

```
import logging
import math

import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score, precision_score, recall_score,
confusion_matrix, roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from src.algorithm.API import API


class Algorithm:
    columns = ["Open", "Close", "High", "Low", "Percent Change"]

    def __init__(self, currency, algorithm, period="1mo", interval="1d"):
```

```
        self.currency = currency
        self.period = period
        self.interval = interval
        self.api = self.__GetApiData()
        self.data = self.api.CountPercentChange()
        self.model = algorithm()

    def __GetApiData(self):
        try:
            result = API(self.currency, period=self.period, interval=self.interval)
        except Exception as e:
            raise ValueError(f"Error getting data from API: {e}")
        return result

    def fetch_and_prepare_data(self, data: pd.DataFrame):
        X = data[self.columns].values
        data['Percent Change Class'] = np.where(data['Percent Change'] > data['Percent
Change'].median(), 1, 0)
        y = data['Percent Change Class'].values
        X = np.array(X).astype(float)
        y = np.array(y).astype(float)
        if len(X.shape) == 1:
            X = X.reshape(-1, 1)
        return X, y

    def train(self):
        logging.info(f"Training the model using {self.model.__class__.__name__}
algorithm",
                     extra={"currency": self.currency})
        X, y = self.fetch_and_prepare_data(self.data)
        self.model.fit(X, y)

    def calculate_volatility(self):
        percent_change = self.data['Percent Change']
        mean_value = percent_change.mean()
        deviation_squares = (percent_change - mean_value) ** 2
        variance = deviation_squares.sum() / len(percent_change)
        standard_deviation = variance ** 0.5
        return standard_deviation

    def classify_volatility(self):
        volatility = self.calculate_volatility()
        if volatility < 0.5:
            return "Low"
        elif volatility < 1:
```

```
            return "Medium"
        else:
            return "High"

    def get_metric(self):
        X, y = self.fetch_and_prepare_data(self.data)
        y_pred = self.model.predict(X)
        accuracy = accuracy_score(y, y_pred)
        precision = precision_score(y, y_pred)
        recall = recall_score(y, y_pred)
        tn, fp, fn, tp = confusion_matrix(y, y_pred).ravel()
        specificity = tn / (tn + fp)
        f1_score = 2 * (precision * recall) / (precision + recall)
        roc_auc_score_ = roc_auc_score(y, y_pred)
        metrics = {
            "Accuracy": accuracy,
            "Precision": precision,
            "Recall": recall,
            "Specificity": specificity,
            "F1 Score": f1_score,
            "Roc Auc Score": roc_auc_score_
        }
        return pd.DataFrame(metrics, index=[1])

    def calculate_standard_deviation(self):
        mean = self.data['Percent Change'].mean()
        number = len(self.data['Percent Change'])
        summa = self.data['Percent Change'].sum()
        formula = lambda x: math.sqrt((summa * (x - mean)) ** 2 / number)
        self.data['Std Deviation'] = self.data['Percent Change'].apply(formula)
        return self.data

    def predict(self):
        data = self.data.copy()
        data['Percent Change Class'] = np.where(data['Percent Change'] > data['Percent
Change'].median(), 'High', 'Low')
        X = data[self.columns].values  # Only use the columns in self.columns for X
        y = data['Percent Change Class'].values
        predictions = self.model.predict(X)
        df = pd.DataFrame(X, columns=self.columns)
        df['Predictions'] = predictions
        return df

    def classify_percent_change(self):
        data = self.data.copy()
```

```
        data['Percent Change Class'] = np.where(data['Percent Change'] > data['Percent
Change'].median(), 'High', 'Low')
        X = data[['Percent Change']].values
        y = data['Percent Change Class'].values
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
        scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.transform(X_test)

        self.model.fit(X_train, y_train)
        latest_data_point = X_test[-1].reshape(1, -1)
        prediction = self.model.predict(latest_data_point)

        return prediction[0]


def test():
from src.algorithm.algorithms import AlgorithmsList
algorithm = Algorithm("UAH=X", algorithm=AlgorithmsList.NaiveBayes)
# Обучение модели
print("Training the model...")
algorithm.train()
#
# Предсказание и вывод результатов
print("Predictions:")
predictions = algorithm.predict()
print(predictions)
#
# Вывод метрик
print("Metrics:")
metrics = algorithm.get_metric()
print(metrics)

if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO)
    test()
```

analyze.py

```
import math
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

from sklearn.model_selection import train_test_split
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report

pd.set_option('display.max_columns', None)

# Завантажте дані з CSV файлу
df = pd.read_csv('UAH=X (1).csv')
# Remove rows with NaN values
df = df.dropna()

df.head()

# Отримайте назви стовпців
column_names = df.columns

# Виведіть назви стовпців
print(column_names)

# Виведіть інформацію про структуру набору даних
print(df.info())

# Перевірка відсутніх значень
missing_values = df.isnull()
print(missing_values)

# Перегляньте перші рядки даних
print(df.head())

# Виведіть останні 5 рядків даних
print(df.tail())

# Виведіть узагальнену інформацію про набір даних
print(df.describe())

# Видаліть стовпці 'Volume' і 'Adj Close'
columns_to_drop = ['Volume', 'Adj Close']
df = df.drop(columns=columns_to_drop, axis=1)

# Виведіть інформацію про структуру набору даних
print(df.info())

# Побудуйте діаграми розподілу для числових стовпців
numeric_columns = df.select_dtypes(include=['float64']).columns

# Встановіть розмір головного графіку
```

```
plt.figure(figsize=(12, 8))

# Додайте підграфіки для кожного числового стовпця
for i, column in enumerate(numeric_columns, 1):
    plt.subplot(2, 2, i)  # 2 рядки, 3 стовпці
    sns.histplot(df[column], kde=True, color='blue')
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')

# Налаштуйте розміщення та відступи між підграфіками
plt.tight_layout()

# Відобразіть графік
plt.show()

# Розрахунок процентного змінення
df['Percent Change'] = (df['Close'] - df['Open']) / df['Open'] * 100

# Розрахунок стандартного відхилення
mean = df['Percent Change'].mean()
number = len(df['Percent Change'])
summa = df['Percent Change'].sum()
formula = lambda x: math.sqrt((summa * (x - mean)) ** 2 / number)
df['Std Deviation'] = df['Percent Change'].apply(formula)

# Виведіть оновлений датафрейм з новим стовпцем "Std Deviation"
print(df.head(10))

# Видаліть стовпець з рядковими значеннями, якщо він є
df_numeric = df.select_dtypes(include=['float64', 'int64'])

# Розрахуйте матрицю кореляції
correlation_matrix = df_numeric.corr()
# Встановіть розмір графіку
plt.figure(figsize=(10, 8))
# Створіть теплокарту кореляції
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
# Встановіть заголовок графіку
plt.title("Correlation Matrix")
# Відобразіть графік
plt.show()

# Розрахунок кореляції
correlation_matrix = df_numeric.corr()
```

```
print("Correlation Matrix:")
print(correlation_matrix)

# Додайте стовпець Volatility Class, де 1 - висока волатильність, 0 - низька
волатильність
volatility_threshold = 0  # Задайте поріг волатильності
df['Volatility Class'] = (df['Percent Change'] > volatility_threshold).astype(int)
# Виведіть оновлений датафрейм
print(df[['Date', 'Percent Change', 'Volatility Class']])
# Встановіть розмір графіку
plt.figure(figsize=(15, 8))
# Великий точковий графік волатильності
sns.scatterplot(data=df, x='Date', y='Percent Change', hue='Volatility Class',
size='Percent Change', sizes=(10, 200),
                palette={0: 'red', 1: 'green'})
plt.title('Volatility Classification Over Time')
plt.xlabel('Date')
plt.ylabel('Percent Change')
plt.legend(title='Volatility Class')
# Відобразіть графік
plt.show()

# Розділіть дані на навчальну та тестову вибірку
train_data, test_data = train_test_split(df, test_size=0.2, random_state=42,
stratify=df['Volatility Class'])

# Виведемо розмір навчальної та тестової вибірки
print(f"Розмір навчальної вибірки: {len(train_data)}")
print(f"Розмір тестової вибірки: {len(test_data)}")

# Вивести розподіл класів в навчальній вибірці
print("Навчальна вибірка:")
print(train_data['Volatility Class'].value_counts())

# Вивести розподіл класів в тестовій вибірці
print("Тестова вибірка:")
print(test_data['Volatility Class'].value_counts())

# Видаліть стовпець 'Date'
df = df.drop(columns=['Date'])

# Розділіть дані на ознаки (X) та цільову змінну (y)
# X = df.drop(columns=['Open', 'High', 'Low', 'Close', 'Percent Change'])
y = df['Volatility Class']
X = df.drop('Volatility Class', axis=1)
```

```
# Розділіть дані на навчальну та тестову вибірку
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

X_train

from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score

# Ініціалізуйте та навчіть модель наївного баєсового класифікатора
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

# Зробіть прогнози для тестового набору даних
y_pred = nb_classifier.predict(X_test)

# Виведення звіту з класифікації
classification_rep = classification_report(y_test, y_pred)
print('Classification Report:')
print(classification_rep)

# Обчислити confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

#  Витягувати значення з confusion matrix
tn, fp, fn, tp = conf_matrix.ravel()

# Обчислити specificity
specificity = tn / (tn + fp)
print('Specificity:', specificity)

# # Обчислити accuracy (точність)
# accuracy = accuracy_score(y_test, y_pred)
# print('Accuracy:', accuracy)

# # Обчислити recall (чутливість)
# recall = recall_score(y_test, y_pred)
# print('Recall:', recall)

# # Обчислити precision (точність)
# precision = precision_score(y_test, y_pred)
# print('Precision:', precision)

# # Обчислити F1 score
```

```
# f1 = f1_score(y_test, y_pred)
# print('F1 Score:', f1)

# Візуалізація confusion matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative',
'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# Visualize other metrics (accuracy, recall, precision, f1)
metrics_names = ['Accuracy', 'Recall', 'Precision', 'F1 Score']
metrics_values = [accuracy_score(y_test, y_pred), recall_score(y_test, y_pred),
precision_score(y_test, y_pred), f1_score(y_test, y_pred)]

plt.figure(figsize=(10, 5))
plt.bar(metrics_names, metrics_values, color=['blue', 'green', 'orange', 'red'])
plt.ylabel('Metric Value')
plt.title('Classification Metrics')
plt.show()

from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Ініціалізуйте та навчіть модель SVM
svm_classifier = SVC()
svm_classifier.fit(X_train, y_train)
# Зробіть прогнози для тестового набору даних
y_pred_svm = svm_classifier.predict(X_test)

# Виведення звіту з класифікації
classification_rep_svm = classification_report(y_test, y_pred_svm)
print('Classification Report (SVM):')
print(classification_rep_svm)

# Обчислити confusion matrix
conf_matrix_svm = confusion_matrix(y_test, y_pred_svm)

# Витягувати значення з confusion matrix
tn_svm, fp_svm, fn_svm, tp_svm = conf_matrix_svm.ravel()
```

```
# Обчислити specificity
specificity_svm = tn_svm / (tn_svm + fp_svm)
print('Specificity (SVM):', specificity_svm)

# Візуалізація confusion matrix для SVM
sns.heatmap(conf_matrix_svm, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative',
'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix (SVM)')
plt.show()

# Visualize other metrics (accuracy, recall, precision, f1) for SVM
metrics_values_svm = [accuracy_score(y_test, y_pred_svm), recall_score(y_test,
y_pred_svm), precision_score(y_test, y_pred_svm), f1_score(y_test, y_pred_svm)]

plt.figure(figsize=(10, 5))
plt.bar(metrics_names, metrics_values_svm, color=['blue', 'green', 'orange', 'red'])
plt.ylabel('Metric Value')
plt.title('Classification Metrics (SVM)')
plt.show()
```

API.py

```
import logging
import pandas
import yfinance as yf

class API:
    def __init__(self,
                 ticker: str,
                 period: str = "1mo",
                 interval: str = "1d"
                 ):
        logging.info(f"Initializing API for {ticker}. Period: {period}, Interval:
{interval}")
        self.ticker = ticker
        self.period = period
        self.interval = interval
        self.data = yf.Ticker(
            ticker=ticker
        )
    def __Update(self) -> yf.Ticker:
        self.data = yf.Ticker(
```

```
            ticker=self.ticker
        )
        return self.data

    def ChangePeriod(self, period: str, interval: str = "1m") -> None:
        self.period = period
        self.interval = interval
        self.__Update()
        return

    def CountPercentChange(self) -> pandas.DataFrame:
        data_df = self.__Update().history(
            period=self.period,
            interval=self.interval
        )
        data_df['Percent Change'] = (data_df['Close'] - data_df['Open']) /
data_df['Open'] * 100
        not_null_columns = data_df[["Open", "Close", "High", "Low", "Percent Change"]]
        not_null_columns = not_null_columns.round(3)
        not_null_columns.index = not_null_columns.index.strftime('%Y-%m-%d %H:%M:%S')
        return not_null_columns

    def GetData(self) -> pandas.DataFrame:
        return self.data.history(
            period=self.period,
            interval=self.interval
        )

def test():
    api = API("UAH=X", period="1d", interval="1m")
    print(api.GetData().to_string(index=True))
    print(api.GetData().to_string(index=True, col_space=12))

if __name__ == "__main__":
    test()
```