

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

(повна назва інституту/факультету)

Інформатики та програмної інженерії

(повна назва кафедри)

«На правах рукопису»

УДК 004.891.2

До захисту допущено:

В.о. завідувача кафедри

_____ Едуард ЖАРІКОВ

«__» _____ 20__ р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інженерія
програмного забезпечення інформаційних систем»**

зі спеціальності 121 «Інженерія програмного забезпечення»

**на тему: «Програмне та математичне забезпечення
автоматизованого трейдингового помічника»**

Виконав (-ла):

студент (-ка) VI курсу, групи IT-04мп

Зайченко Василь Володимирович _____

Науковий керівник:

к.т.н., доцент

Фіногенов Олексій Дмитрович _____

Рецензент:

д.т.н., доцент

Корнага Ярослав Ігорович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент (-ка) _____

Київ – 2021 рік

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма - «Інженерія програмного забезпечення інформаційних систем»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Едуард ЖАРІКОВ

«___» _____ 20__ р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Зайченку Василю Володимировичу

1. Тема дисертації: «Програмне та математичне забезпечення автоматизованого трейдингового помічника», науковий керівник дисертації Фіногенов Олексій Дмитрович, доцент кафедри інформатики та програмної інженерії, затверджені наказом по університету від «26» жовтня 2021 р. №3132-с
2. Термін подання студентом дисертації: 16.12.2021 _____
3. Об'єкт дослідження: процес формування стабільного фінансового портфелю.
4. Вхідні дані: постановка задачі _____
5. Перелік завдань, які потрібно розробити: мікросервісне програмне забезпечення автоматизованого торгового помічника; модель прогнозування біржових цін акцій компанії на основі галузі.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
 - діаграма класів;
 - схема бази даних;
 - структурна схема архітектури програмного забезпечення;
 - лістинг програми;
 - перевірка на антиплагіат.

7. Перелік публікацій: Зайченко В.В. «Аналіз методів часових рядів на основі коливань волатильності компаній німецького ринку».

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Графічний	доц. Фіногенов О.Д.		

9. Дата видачі завдання 1.09.2021

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вивчення рекомендованої літератури	01.09.2021	
2	Аналіз існуючих методів розв'язання задачі	10.09.2021	
3	Постановка та формалізація задачі	15.09.2021	
4	Моделювання ПЗ	24.09.2021	
5	Визначення з датою сетами	27.09.2021	
6	Аналіз та реалізація існуючих алгоритмів і дослідження їх	4.10.2021	
7	Розробка архітектури ПЗ	01.11.2021	
8	Розробка і налагодження програмного забезпечення	08.11.2021	
9	Оформлення пояснювальної записки	20.11.2021	
10	Подання МД на попередній захист	23.11.2021	
11	Подання МД рецензенту	10.12.2021	
12	Подання МД на основний захист	22.12.2021	

Студент

Василь ЗАЙЧЕНКО

Науковий керівник

Олексій ФІНОГЕНОВ

РЕФЕРАТ

Розмір пояснювальної записки – 85 аркушів, містить 39 ілюстрації, 29 таблиць, 5 додатків. Програмне та математичне забезпечення автоматизованого трейдингового помічника.

Актуальність теми: автоматичні експерти та роботи для трейдингу керуються інструментами технічного та математичного аналізу, проте вони не враховують фундаментальної інформації торгівлі та різкої мінливості ринку. Якісні радники дуже дорогі і мають обмежений термін придатності. Стратегії фінансового ринку з плином часу втрачають власну актуальність та перестають бути прибутковими для користувача.

Мета дослідження: вдосконалення програмного забезпечення прогнозування біржових цін акцій компаній використовуючи мікросервісну архітектуру на основі галузей та індексів. Для досягнення мети дослідження поставлено і вирішено такі **завдання:**

- аналіз сучасних рішень та методів;
- розробка мікросервісного програмного забезпечення автоматизованого торгового помічника;
- прогнозування біржових цін акцій компаній на основі галузей;
- аналіз отриманих результатів у експериментальних дослідженнях.

Об'єкт дослідження: процес формування стабільного фінансового портфелю.

Предмет дослідження: методи та програмні засоби для прогнозування довготермінового курсу біржових цін акцій.

Наукова новизна: удосконалення програмного забезпечення автоматизованого торгового помічника на основі мікросервісної архітектури для паралельного прогнозування цін акцій для галузевих та індексних компаній.

Практичне значення: веб-додаток з прогнозованими довготерміновими біржовими цінами акцій. Реалізовано програмний

інтерфейс для отримання та використання результатів моделі сторонніми додатками.

Зв'язок з науковими програмами, планами, темами: робота виконувалась в межах теми «Методи та технології високопродуктивних обчислень та обробки надвеликих масивів даних», державний реєстраційний номер 0117U000924 на кафедрі інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Публікації: «Аналіз методів часових рядів на основі коливань волатильності компаній німецького ринку» в науковому журналі «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки» у Томі 32 (71) № 6 за 2021.

Ключові слова: ТОРГОВИЙ ПОМІЧНИК, МІКРОСЕРВІСНА АРХІТЕКТУРА, РЕГРЕСІЙНИЙ АНАЛІЗ, ЧАСОВИЙ РЯД, ДОВГОТЕРМІНОВЕ ПРОГНОЗУВАННЯ.

ABSTRACT

Explanatory note size – 85 pages, contains 39 illustrations, 29 tables, 5 applications. Software and mathematical software of automated trading advisor.

Topicality: automatic experts and robots are guided by the tools of technical and mathematical analysis, but they do not take into account the fundamental information of trade and the sharp variability of the market. Quality advisors are very expensive and have a limited shelf life. Over time, financial market strategies lose their relevance and cease to be profitable for the user.

The aim of the study: the purpose of the master's thesis is to improve the software for forecasting stock prices of companies using microservice architecture based on industries and indices. To achieve the goal of the study, **the following tasks** were set and solved:

- analysis of modern solutions and methods;
- development of microservice software for automated sales assistant;
- forecasting stock exchange prices of companies based on industries;
- analysis of the results obtained in experimental studies.

Object of research: the process of forming a stable financial portfolio.

Subject of research: methods and software for forecasting the long-term price of stock prices.

The scientific novelty: the improvement of the software of the automated trading assistant on the basis of microservice architecture for parallel forecasting of stock prices for branch and index companies.

The practical value: web application with projected long-term stock market prices. Implemented a software interface for obtaining and using the results of the model by third-party applications.

Relationship with working with scientific programs, plans, topics: the work was performed within the theme «Methods and technologies of high-performance computing and processing of ultra-large data sets», state registration number 0117U000924 at the Department of Informatics and Software Engineering

of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

Publications: «Analysis of time series methods based on fluctuation volatility of German market companies» was adopted by the scientific journal «Scientific Notes of the Tavriya National University named after VI Vernadsky». Series: Technical Sciences and will be published in Volume 32 (71) № 6 for 2021.

Keywords: TRADING ASSISTANT, MICROSERVICE ARCHITECTURE, REGRESSION ANALYSIS, TIME SERIES, LONG-TERM FORECASTING.

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	7
1.1 Аналіз предметної області	7
1.2 Порівняння алгоритмів прогнозування	10
1.3 Аналіз існуючих рішень	12
1.3.1 Сервіс «StrategyQuant»	12
1.3.2 Сервіс «Forex Trendy Scanner»	14
1.3.3 Сервіс «EA Builder»	16
1.3.4 Порівняння аналогів	18
1.4 Висновки до розділу	20
2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	21
2.1 Загальні положення про гауссову модель процесів	21
2.2 Багатовимірний нормальний розподіл	24
2.3 Функції ядра Гаусса	27
2.4 Непараметричні моделі	29
2.5 Процес Гаусса	30
2.6 Обробка новин	34
2.7 Оптимізація гіперпараметрів	38
2.8 Висновки до розділу	42
3 СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	43
3.1 Аналіз програмного забезпечення	43
3.2 Архітектура програмного забезпечення	44
3.2.1 Головні класи програмного забезпечення	44
3.2.2 Кінцеві точки зв'язку	46
3.3 Аналіз безпеки даних користувача	51
3.4 Інструкція для користувача та демонстрація проекту	51
3.5 Генерація датасету	57
3.6 Експериментальні дослідження	59
3.7 Оцінка ефективності запропонованого рішення	63

	3
3.8 Висновки до розділу	66
4 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ	68
4.1 Опис ідеї проєкту	68
4.2 Технологічний аудит ідеї проєкту	68
4.3 Аналіз ринкових можливостей запуску стартап-проєкту	71
4.4 Розроблення ринкової стратегії проєкту	77
4.5 Розроблення маркетингової програми стартап-проєкту	80
4.6 Висновки по розділу	83
ВИСНОВКИ	84
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	86
ДОДАТКИ	
ДОДАТОК А	
ДОДАТОК Б	
ДОДАТОК В	
ДОДАТОК Г	
ДОДАТОК Д	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

10-K – річний звіт про фінансові результати біржової компанії.

10-Q – квартальний звіт про фінансові результати біржової компанії.

A (Agilent Technologies) – продовольча компанія, яка котирується на NYSE.

AAP (Advance Auto Parts) – автомобільна компанія, яка котирується на NYSE.

CART (дерево класифікації та регресії) – модель для прогнозування значень змінної результату на основі інших значень.

EA – експертний радник.

Endpoint – інтерфейс для взаємодії клієнтською частини з серверною.

IBM (International Business Machines) – технологічна компанія, яка котирується на NYSE.

KEY – індустріальна компанія, яка котирується на NYSE.

NASDAQ – це глобальна онлайнова біржа для торгівлі та купівлі цінних паперів, перша у світі електронна біржа.

NYSE (Нью-Йоркська фондова біржа) – найбільша фондова біржа США.

S&P 500 – фондовий індекс, до кошика якого входять 500 акцій обраних публічних компаній, що мають найбільшу капіталізацію, які котируються на фондових біржах США.

SVM (метод опорних векторів) – набір подібних алгоритмів навчання, що використовуються для вирішення завдань класифікації та регресії.

UPS (United Parcel Service) – логістична компанія, яка котирується на NYSE.

БНР – багатовимірний нормальний розподіл.

ГМП – гауссова модель процесів.

ДНР – двовимірний нормальний розподіл.

КЕФ – квадратна експоненціальна функція.

РБФ – радіальної базисної функції.

ФЙЩ – функція ймовірності щільності.

ВСТУП

Прогнозування тенденцій цін на акції є активним напрямком дослідження, оскільки точніші прогнози безпосередньо пов'язані з більшою прибутковістю акцій. Тому в останні роки були докладені значні зусилля для розробки моделей, які можуть передбачити майбутні тенденції певної акції або ринку в цілому. Дослідники показали, що існує сильний зв'язок між новиною про компанію та коливаннями її цін на акції. Нижче наведено обговорення попередніх досліджень щодо аналізу настроїв текстових даних та різних методів класифікації та регресії. Нагар і Хаслер у своєму дослідженні [7] представили підхід, заснований на автоматизованому аналізі тексту, до агрегування новин з різних джерел і створення корпусу новин. Корпус фільтрується до відповідних речень і аналізується за допомогою методів обробки природної мови (NLP).

Ю та інші [26] представляють структуру на основі аналізу тексту, щоб визначити настрої новинних статей та проілюструвати їх вплив на попит на енергію. Настрої новин визначають кількісно, а потім представляють у вигляді часового ряду та порівнюють із коливаннями попиту на енергію та цін. Дж. Бін [2] використовує тегування ключових слів у каналах Twitter про задоволеність авіакомпаній, щоб оцінити їх за полярність і настрої. Це може дати швидке уявлення про настрої, що панують щодо авіакомпаній, та їх рейтинги задоволеності клієнтів.

У наступній дослідницькій роботі [6] досліджується, як можна покращити результати фінансового прогнозування при одночасному використанні новинних статей з різним рівнем релевантності цільовому фонду. Результати експерименту показують, що одночасне використання п'яти категорій новин покращує ефективність прогнозування порівняно з методами, заснованими на меншій кількості категорій новин.

Автоматичні експерти та роботи керуються інструментами технічного та математичного аналізу, проте вони не враховують фундаментальної інформації торгівлі та різкої мінливості ринку. Якісні радники дуже дорогі і

мають обмежений термін придатності. Будь-яка стратегія з плином часу втрачає власну актуальність та перестає бути прибутковою для користувача.

Мета дослідження: вдосконалення програмного забезпечення прогнозування біржових цін акцій компаній використовуючи мікросервісну архітектуру на основі галузей та індексів. Для досягнення мети дослідження поставлено і вирішено такі **завдання:**

- аналіз сучасних рішень та методів;
- розробка мікросервісного програмного забезпечення автоматизованого торгового помічника;
- прогнозування біржових цін акцій компаній на основі галузей;
- аналіз отриманих результатів у експериментальних дослідженнях.

Об’єкт дослідження: процес формування стабільного фінансового портфелю.

Предмет дослідження: методи та програмні засоби для прогнозування довготермінового курсу біржових цін акцій.

Наукова новизна: удосконалення програмного забезпечення автоматизованого торгового помічника на основі мікросервісної архітектури для паралельного прогнозування цін акцій для галузевих та індексних компаній.

Практичне значення: веб-додаток з прогнозованими довготерміновими біржовими цінами акцій. Реалізовано програмний інтерфейс для отримання та використання результатів моделі сторонніми додатками.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз предметної області

Експертний радник (ЕА) — це частина програмного забезпечення, яка приймає торгові рішення від вашого імені. Після встановлення Forex EA на торговій платформі експерт-радник або торговий робот керуватиме вашими угодами на основі деяких заздалегідь визначених критеріїв. Ці критерії є певною формою запрограмованих алгоритмів на основі вказаних вами торгових стратегій [26].

Перший робот у цьому списку форекс-роботів - це той, який торгує найпопулярнішою валютною парою на ринку – EURUSD. Дані валюти люблять більшість трейдерів через їх високу волатильність, що полегшує отримання прибутку. Підраховано, що на валютну пару припадає близько 25% щоденного обсягу торгівлі на валютному ринку. Через великий обсяг торгів і величезний попит, EURUSD має найнижчі спреди на ринку. Крім того, він представляє валюти двох основних економік світу, тобто єврозони та США. Долар є світовою резервною валютою, тоді як євро є однією з найважливіших валют світу. Через ці переваги більшість ЕА створено для торгівлі EURUSD. Експерти для EURUSD зазвичай розробляються, щоб гарантувати, що трейдери отримують максимальну віддачу від торгівлі валютною парою. Ці ЕА створені з урахуванням характеристик валютної пари, що гарантує, що трейдери можуть отримувати хороші доходи за допомогою роботів. Зазвичай на EURUSD найбільше впливають новини, такі як оголошення про зарплату в несільськогосподарських підприємствах та оголошення про процентні ставки. Це означає, що дана пара багато рухається, і це ідеальна пара для торгівлі [29].

Робот Forex на основі ковзних середніх широко застосовується у аналітиці. Торгівля за допомогою ковзних середніх є однією з широко використовуваних стратегій на ринку форекс. Оскільки ковзні середні є індикаторами, які згладжують коливання валютних пар протягом певного

періоду часу, вони полегшують визначення торгових можливостей. У торгівлі на фореक्स ковзні середні можна використовувати різними способами, включаючи визначення ринкової тенденції, відмічення значних рівнів підтримки та опору, а також виконання угод на основі їх кросовера [23]. Наприклад, якщо стратегія заснована на перетинаннях ковзних середніх, сигнал на покупку може бути вироблений щоразу, коли короткострокова ковзна середня перетинає довгострокову ковзну середню. Робот Форекс, заснований на ковзних середніх, зазвичай користується перевагами різних способів використання індикаторів і заробляння грошей. Ви також можете створити робота на основі стратегії, яка поєднує в собі кілька показників і залежить від ковзних середніх як додаткових критеріїв для підтвердження точок входу та виходу на ринок. За допомогою такої комбінації кількість помилок може бути значно зменшена через безліч умов, які необхідно виконати перед виконанням угоди [27].

Мікрорахунок з капіталом менше \$ 100 і дозволяє торгувати з 0,01 мікролотів. З мікрорахунком можливо безпечно випробовувати свої стратегії в умовах реальної торгівлі, не турбуючись про величезні втрати. Аналогічно, експертний радник з мікрорахунків дає можливість випробувати робота на невеликому торговому рахунку, перш ніж перейти на більший рахунок [25]. Незважаючи на те, що ЕА для невеликих облікових записів важко підтримувати, а підтримувати належне управління грошима під час їх використання непросто, вони все ще існують і можуть принести хорошу віддачу особливо для новачків.

Скальпінг - це торгова техніка, яка передбачає відкриття кількох позицій на ринку для короткострокового прибутку. Намір скальпера полягає в тому, щоб скористатися перевагами невеликих ринкових рухів, щоб здійснити численні операції, які в кінцевому підсумку підсумовуються в кінці дня. Експерт для скальпінгу створений, щоб допомогти трейдерам отримати виграш від короткострокових ринкових рухів [24]. Скальпуючий ЕА зазвичай забирає на ринку короткострокові торгові можливості. Незважаючи на те, що

форекс-роботи для скальпінгу популярні, оскільки більшість людей люблять швидкі способи отримання прибутку, вони зазвичай є великою марною тратою грошей і можуть не виявитися стійкими в довгостроковій перспективі. Ринок форекс зазвичай динамічний за своєю природою — непередбачувані промахи, спреди та інші вразливості можуть призвести до великих витрат на торгівлю. Оскільки скальпінг ЕА запрограмований на швидке реагування на короточасні ринкові умови, усунути такі вразливості може бути важко. Крім того, більшість скальпінгових експертів зазвичай закодовані для роботи лише з одним певним типом брокерського рахунку. Тому використання їх для інших рахунків іншими брокерами може бути неможливим. Замість того, щоб витрачати багато грошей на скальпінг-роботів, які можуть не дати вам гарної віддачі, краще інвестувати в інструменти ЕА-генератора, як-от Forex Robot Factory, щоб отримати необмежену кількість експертів-порадників для торгівлі на форекс, які використовують середньострокові і довгострокові торгові стратегії [22].

Торгівля на основі новин люблять більшість трейдерів через її здатність отримувати значний прибуток протягом короткого періоду часу. Щоразу, коли робиться важливе економічне оголошення, ринок зазвичай реагує і стрибає вгору або вниз. Ще один ЕА у списку експертів – це експертний радник для торгівлі новинами, який зазвичай створюється з параметрами, призначеними для прийняття найкращих торгових рішень під час економічних анонсів. З таким роботом ви можете скористатися високою волатильністю під час випусків новин і збільшити свій торговий прибуток. Під час економічних випусків ситуація на валютному ринку може змінюватися дуже швидко. Тому для того, щоб такі ЕА працювали добре, їх слід запрограмувати, щоб пристосуватись до величезних проступів, розширених спредів і навіть реквотування цін [22]. Це не має значення, чи створений торговий робот для роботи на більш високих часових кадрах, як-от 4-годинний або щоденний. Крім того, щоб експертний радник з торгівлі новинами був прибутковим, він повинен враховувати управління ризиками. Якщо ви станете занадто

жадібними і не зможете закріпити критерії управління ризиками у своєму ЕА, то великі кроки під час важливих ринкових оголошень можуть працювати проти вас.

1.2 Порівняння алгоритмів прогнозування

Для порівняння алгоритмів були взяті компанії, які котируються на американській біржі NYSE. У середньому для кожної компанії доступно акційні ціни протягом 10 років (приблизно 3000 денних значень). Для прогнозування ціни закриття були використано такі дані:

- ціна відкриття (open price);
- ціна закриття (close price).

Було взято декілька періодів для уточнення та обчислення ефективності прогнозування моделей:

- 750 днів (2 торгових роки);
- 1500 днів (5 торгових років);
- 2000 (7 торгових років).

Характеристики оцінювання алгоритмів:

- точність моделі поточного дня - це процент компаній, у яких насправді змінилася ціна на 5% в останній (на момент запуску та прогнозування) день, наприклад, якщо це значення дорівнює 80% і модель видала список з 10 компаній, які мають змінитися на 5%, то у реальності ціна змінилася у 8 компаній з прогнозованих 10;

- процент зміни портфелю - це зміна вартості портфеля протягом 5 торгових днів;

- середнє значення моделі для компанії - це середнє значення усіх прогонів моделі (для кожної компанії модель окремо перенавчається), наприклад, якщо це значення також дорівнює 80%, і тестова вибірка складає 100 записів цін однієї певної компанії, то 80 разів модель спрогнозує правильно.

Таблиця 1.1 — Порівняння алгоритмів з періодом 750 днів

Алгоритм	Точність моделі поточного дня	Процент зміни портфелю	Середнє значення моделі для компанії
LSTM	75.96%	0.34%	78.68%
ARIMA	70.48%	4.60%	75.73%
SARIMA	78.37%	4.65%	76.19%
SVM	70.48%	-1.43%	76.17%
Gauss	77.80%	1.17%	85.92%

Таблиця 1.2 — Порівняння алгоритмів з періодом 1500 днів

Алгоритм	Точність моделі поточного дня	Процент зміни портфелю	Середнє значення моделі для компанії
LSTM	72.35%	0.145%	76.94%
ARIMA	70%	0.98%	77.32%
SARIMA	75%	1.73%	79.20%
SVM	70.35%	2.77%	79.45%
Gauss	80.15%	1.18%	77.99%

Таблиця 1.3 — Порівняння алгоритмів з періодом 2000 днів

Алгоритм	Точність моделі поточного дня	Процент зміни портфелю	Середнє значення моделі для компанії
LSTM	71.63%	0.286%	70.94%
ARIMA	77.73%%	2.76%	70.91%
SARIMA	70.35%	-1.67%	68.34%
SVM	78.89%	1.59%	79.44%
Gauss	71.67%	1.64%	78.19%

«StrategyQuant» надає інструменти професійних квантів і хедж-фондів для створення портфеля торгових стратегій кількісним способом. Основні характеристики «StrategyQuant»:

- автоматизована побудова стратегії без навичок програмування;
- підтримує платформи MetaTrader, TradeStation і NinjaTrader;
- включає в себе більше 40 індикаторів, патернів, свічників і т. д;
- 4 режими (побудова, повторне тестування, поліпшення, оптимізація);
- сотні існуючих стратегій, що охоплюють кожен стиль торгівлі;
- 4 ордери на вхід, 7 ордерів на вихід і інтелектуальний стоп-лосс;
- автоматизоване тестування на основі широкого спектру параметрів;
- використання випадковості для тестування стратегії в будь-яких ринкових умовах;
- покрокова оптимізація;
- побудова і повторне тестування стратегій для іншого ринку або таймфрейма;
- широкі можливості для фільтрації стратегій.

«StrategyQuant» пропонує трейдерам можливість генерувати і тестувати тисячі різних умов входу і виходу, типів ордерів і рівнів цін, щоб знаходити найбільш ефективні автоматичні стратегії відповідно до спеціальних критеріїв, наприклад, прибутковість проти просадки, коефіцієнт Шарпа і т.д.

Переваги та недоліки «StrategyQuant»:

- розробка стратегії за допомогою дуже зручного інтерфейсу і сотень вбудованих стратегій;
- чотири режими (побудова, повторне тестування, поліпшення, оптимізація);
- експортовані файли можуть використовуватися на декількох платформах;
- включає в себе більше 40 індикаторів, патернів, свічників і т. д;

- автоматизоване тестування з використанням випадковості і на основі широкого спектру параметрів;
- вбудований оптимізатор руху вперед плюс безліч інших інструментів оптимізації (Монте-Карло і т. д.);
- базова ліцензія коштує 1290 доларів США;
- стратегія залежить лише від технічного аналізу;
- залежить від мінливості ринку;
- немає індикаторів профіту.

1.3.2 Сервіс «Forex Trendy Scanner»

«Forex Trendy» - це рішення для тих трейдерів, які зосереджені на кращому часу для торгівлі на ринку Форекс. Без використання індикаторів для визначення основного тренда, заснований на ринкових діях. «Forex Trendy» здатний сканувати 34 валютні пари на декількох тимчасових інтервалах, які варіюються від 1 хвилини до 1 місяця. Даний сервіс здатний визначати основний тренд, визначаючи нові графічні патерни, коли вони виникають. Він миттєво відстежує всі доступні часові рамки і розпізнає прориви патернів. Це відбувається, коли генеруються звукові оповіщення. Система може бути адаптована до більшості популярних торгових платформ, таких як MetaTrader, TradeStation і NinjaTrader.



Рисунок 1.2 — Приклад роботи сервісу «Forex Trendy Scanner»

Що входить в пакет «Forex Trendy»:

- графіки популярних валютних пар в декількох часових періодах;
- автоматичний аналіз графіків з визначенням прапорів, трикутників, ліній тренда і клинів по 34 валютним парам;
- оповіщення (звукові і по електронній пошті);
- короткий огляд всіх тенденцій і всіх валютних пар;
- часові рамки або валютні пари з автоматично згенерованого аналізу.

«Forex Trendy» заснований на складному алгоритмі, коли справа доходить до виявлення нових тенденцій. Ці тенденції генеруються виключно дією ринку, а не будь-яким індикатором. Це досить незвично для торгової системи в даний час, але це може бути перевагою в короткострокових часових рамках. Як уже згадувалося, «Forex trendy» використовує 9 часових рамок (від 1-хвилинного до 1-місячного графіка), в той час як загальна кількість аналізованих валют становить 34. Це означає, що алгоритм аналізує 306 ($9 * 34$) загальних діаграм.

Команда «Forex Trendy» використовує потужні комп'ютери та сервери, щоб результати їх алгоритму були представлені миттєво

«Forex Trendy» надає повний доступ до зони учасників, де трейдери можуть насолоджуватися кращими рушіями валют на 9 таймфреймах. Крім того, за допомогою аналізу графіків пропонується автоматизована ідентифікація популярних моделей (в основному прапорів, трикутників і ліній тренда). Ось особливості «Forex Trendy»:

- щоденний аналіз 34 валютних пар (9 таймфреймів);
- сигнали по електронній пошті плюс звукові оповіщення;
- просте налаштування валютної пари (1 клік);
- візуалізований торговий інтерфейс;
- автоматичне виявлення прапорів, ліній тренду, трикутників і клинів;
- безкоштовна електронна книга (30 сторінок) про аналіз діаграм;

- ліцензія з повним функціоналом коштує 564 доларів США;
- стратегія залежить лише від технічного аналізу;
- залежить від мінливості ринку;
- немає індикаторів профіту.

1.3.3 Сервіс «EA Builder»

«EA Builder» - це веб-додаток, який допомагає трейдерам перетворювати свої торгові ідеї в повністю автоматизовані системи. Немає необхідності в навичках програмування, і навіть новачки можуть створювати складні радники. «EA Builder» також може створювати прості індикатори, і ця послуга на 100% безкоштовна. Можливо створити автоматичну стратегію для торгівлі будь-якими фінансовими активами (форекс, акціями, товарами) і використовувати її на трьох торгових платформах.



Рисунок 1.3 — Приклад роботи сервісу «EA Builder»

«EA Builder» пропонує зручний інтерфейс з графічними функціями і спливаючими вікнами сповіщень. Є також 15 відеоуроків для подальшого користування. «EA Builder» пропонує повний набір функцій, включаючи розрахунок стоп-лосс і більш складні інструменти управління капіталом.

Конструктор радників безкоштовний для створення індикаторів і коштує 97 доларів США за створення радників.

Це основні функції конструктора радників на перший погляд:

- безкоштовно для створення індикаторів для MT4, MT5 і TradeStation, оплачується тільки за створення складних систем;
- стовідсотковий веб-додаток, що не вимагає навичок програмування;
- радники «EA Builder» можуть торгувати на будь-якому фінансовому ринку (Форекс, акції та товари);
- радники EA Builder можуть торгувати будь-яким активом на будь-якому таймфреймі; торгівля бінарними опціонами також включена;
- високо налаштовується побудова радника (десятки функцій); служба оповіщення по електронній пошті, аудіо і на екрані;
- повний контроль управління капіталом (контроль спреда і прослизання, чутливі ордера);
- «EA Builder» також пропонує торгівлю бінарними опціонами;
- може використовуватися на необмежених реальних або демо-рахунках.

«EA Builder» пропонує велику різноманітність стандартних індикаторів, а також повний набір вбудованих функцій. Ці функції включають підтримку і опір, лінії тренда, тимчасові переваги і багато іншого:

- повна торгова функціональність;
- створення індикаторів для безкоштовних і призначених для користувача радників;
- стрілки за допомогою графічного інтерфейсу;
- необмежені змінні для створення автоматичних стратегій;
- повні функції управління капіталом;
- кольоровий індикатор CCI і попередження про розрив тренда.

«EA Builder» – це веб-додаток, тому не потрібно нічого встановлювати на свій комп'ютер. Існує безліч спливаючих підказок, які призначені для

початківців. «EA Builder» сумісний з усіма браузерами і може працювати на будь-якому комп'ютері (ПК, Mac або Linux). Створення експертів демонструється в MetaTrader, але також і в TradeStation. Можливо отримати покрокову допомогу для створення надійної автоматичної стратегії Форекс. EA Builder-дуже цікавий додаток, який можна безкоштовно використовувати при створенні індикаторів. Ось деякі переваги:

- створення індикаторів безкоштовно і без будь-яких навичок програмування;
- «EA Builder» включає в себе десятки функцій, які можуть інтегрувати ваші ідеї і перетворити їх в повністю працюючу автоматизовану систему Форекс;
- працює на будь-якому комп'ютері або смартфоні;
- згенеровані радники можуть працювати на різних платформах;
- можна створювати радників, які торгують на будь-якому фінансовому ринку за допомогою будь-якого інструменту (включаючи бінарні опціони) ;
- ліцензія з повним функціоналом коштує 564 доларів США.
- стратегія залежить лише від технічного аналізу;
- залежить від мінливості ринку;
- немає індикаторів профіту;
- жорсткий контроль з боку керівництва (розмір, ризик, поширення і час).

1.3.4 Порівняння аналогів

Для порівняння найпопулярніших аналогів автоматизованого трейдингового помічника, а саме «StrategyQuant», «Forex Trendy Scanner» та «EA Builder», було взято такі критерії:

Таблиця 1.4 — Порівняння аналогів

Критерії оцінювання	«StrategyQuant»	«Forex Trendy Scanner»	«EA Builder»
Безкоштовна ліцензія	-	+-	+-
Інтеграція з торговими платформами	+	+	+
Мікросервісне архітектурне рішення	-	-	-
Зручний веб-інтерфейс	+	-	-
Індексне прогнозування	+	-	+
Прогнозування по галузям	-	+	+
Використання фундаментальної інформації ринку	-	-	-

Порівнюючи аналоги видно, що кожне реалізоване програмне рішення має недоліки. Незважаючи на переваги, які реалізують додатки, жоден з них не надає повну або часткову функціональність для таких критеріїв:

- мікросервісне архітектурне рішення;
- використання фундаментальної інформації ринку;
- безкоштовна ліцензія.

1.4 Висновки до розділу

Під час аналізу існуючих методів та аналогів на ринку було сформовано вимоги до програмного забезпечення, яке розроблене у рамках магістерської дисертації:

- зменшення ризиків користувачів;
- покращення ефективності прогнозування цін акцій;
- оптимізація торгового портфелю;
- створення графічного інтерфейсу для отримання результатів;
- розробка програмного інтерфейсу для отримання та використання результатів моделі сторонніми додатками;
- автоматизація користувача.

У першому розділі було:

- сформульовано завдання і цілі дослідження;
- проаналізовано предметну область проблематики;
- сформульовані функціональні вимоги для математичних і програмних забезпечень;
- проаналізовано існуючі методи та аналоги даного рішення.

2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Загальні положення про гауссову модель процесів

Гауссова модель процесів (далі - ГМП) - це імовірнісна система машинного навчання, яка широко використовується для задач регресії та класифікації [1]. Випадкова величина X є гауссова або нормально розподіленою із середнім значенням μ та дисперсією σ , якщо її функція ймовірності щільності (ФЙЩ) дорівнює [10]

$$P_x(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-1(x - \mu)^2}{2\sigma^2}\right)$$

У регресії, враховуючи деякі спостережувані точки даних, підігнано функцію для представлення цих точок даних, а потім використано функцію для прогнозування нових точок даних. Для даного набору спостережуваних точок даних, показаних на рисунку 2.1, існує нескінченна кількість можливих функцій, які відповідають цим точкам даних.

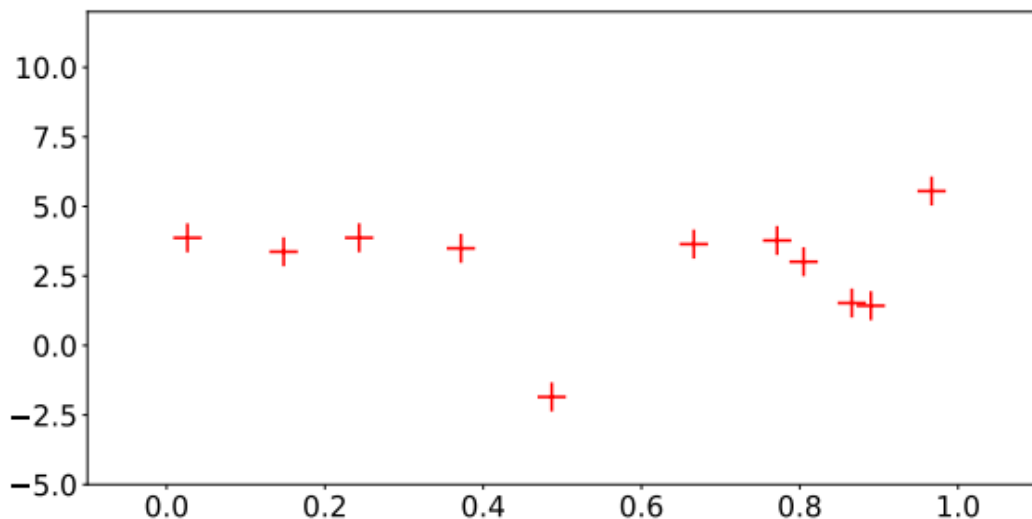


Рисунок 2.1 — Приклади регресії спостережуваних точок даних

На рисунку 2.2 показано п'ять зразкових функцій, які відповідають наборам точок даних. У ГМП гауссові процеси здійснюють регресію, визначаючи розподіл за цим нескінченним числом функцій [9].

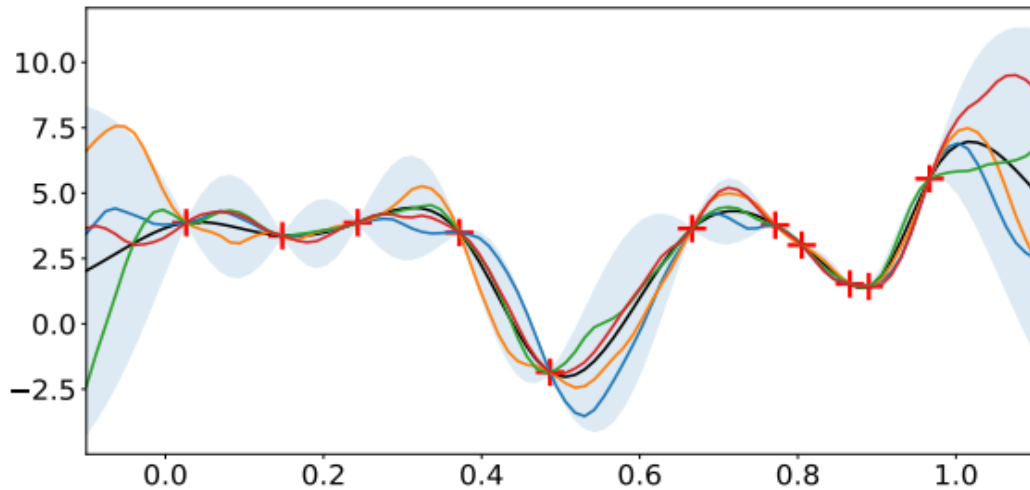


Рисунок 2.2 — Функції вибірки, які відповідають точкам даних

Тут X представляє випадкові величини, а x - реальний аргумент. Нормальний розподіл X зазвичай представлений $P_X(x) \sim N(\mu, \sigma)$. ФЙЩ одномірного нормального (або гауссівського) розподілу було нанесено на рисунку 2.2. Випадковим чином згенеровано 1000 точок з однорідного нормального розподілу та наносено їх на вісь x .

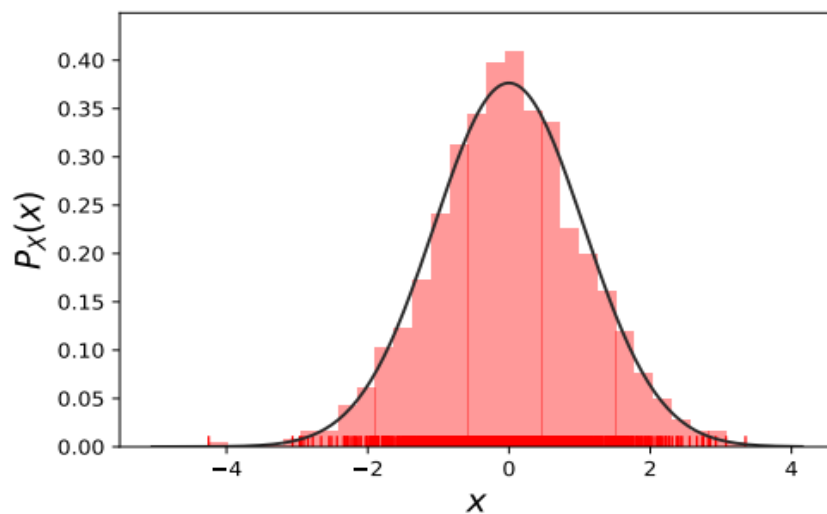


Рисунок 2.3 — Розподілення згенерованих точок

Ці випадково згенеровані точки даних можна виразити у вигляді вектора $x_1 = [x_1^1, x_1^2, x_1^3, \dots, x_1^n]$. Побудувавши вектор x_1 на новій осі Y при $Y = 0$, спроектуємо дані точки в інший простір, показаний на рисунку 2.3. Можемо побудувати інший незалежний вектор Гаусса $x_2 = [x_2^1, x_2^2, x_2^3, \dots, x_2^n]$ в тих самих координатах при $Y = 1$, показаних на рисунку 2.4.

Далі випадковим чином вибрано 10 точок у векторах x_1 і x_2 відповідно і з'єднано ці 10 точок у порядку лініями, як показано на рисунку 2.3. Ці зв'язані лінії виглядають як лінійні функції, що охоплюють область $[0, 1]$. Можемо використовувати ці функції, щоб робити прогнози для завдань регресії, якщо нові точки даних знаходяться на цих лінійних лініях (або досить близько до них) [11]. Однак у більшості випадків не виконується припущення, що нові точки даних завжди знаходяться на зв'язаних лінійних функціях.

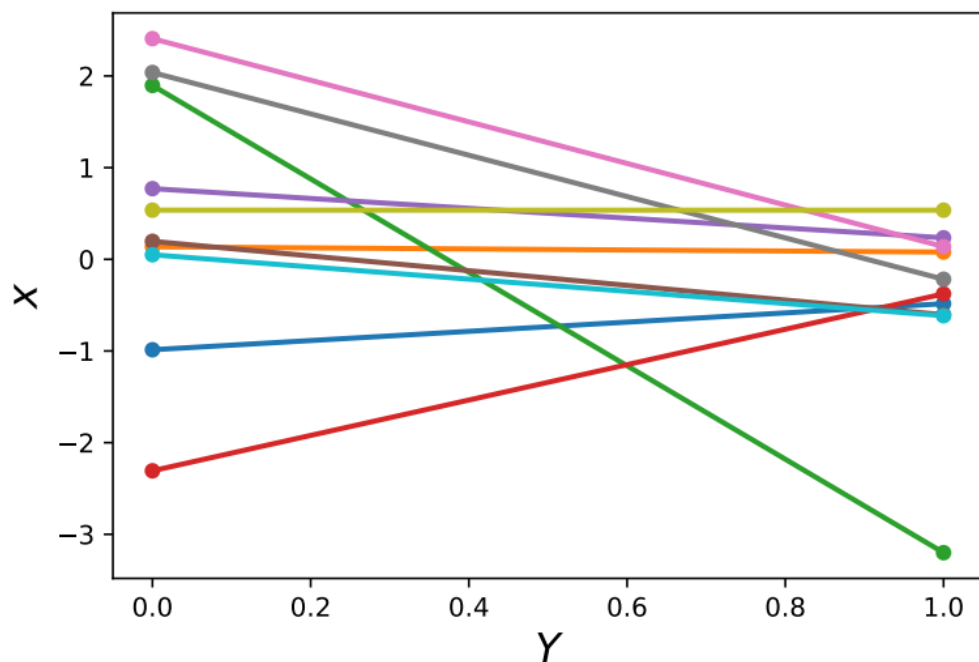


Рисунок 2.4 — Лінійні функції десяти випадково вибраних точок двох векторів

Якщо побудувати більше випадково згенерованих одновимірних гауссівських векторів, наприклад, 20 векторів $x_1, x_2, \dots, x_{20}$ в $[0, 1]$ і з'єднати

10 випадково вибраних точок вибірки кожного вектора як лінії, отримаємо 10 ліній, які більше схожі на функції в межах $[0, 1]$, показані на рисунку 2.5.

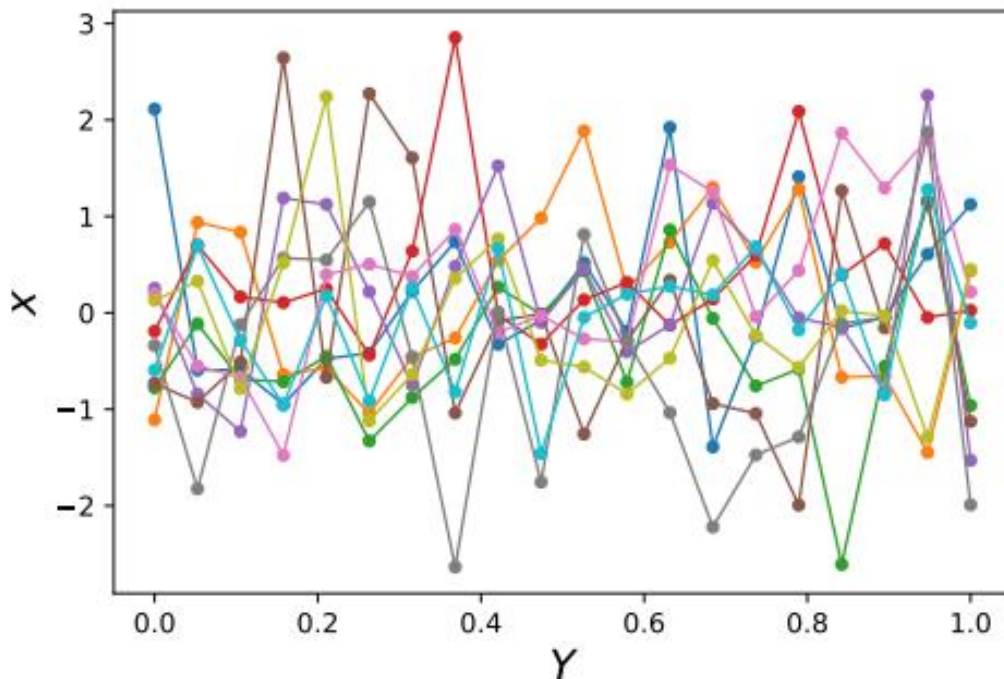


Рисунок 2.5 — Лінійні функції двадцяти векторів

Неможливо використовувати ці рядки для прогнозування завдань регресії, оскільки вони занадто шумні [11]. Ці функції повинні бути більш гладкими, тобто точки входу, які розташовані близько один до одного, повинні мати подібні вихідні значення. Функції, які генеруються поєднанням незалежних гауссівських векторних точок, недостатньо гладкі для завдань регресії, потрібно ці незалежні гауссові вектори скорелювати один з одним як спільний гауссовий розподіл. Спільний гауссівський розподіл описується багатовимірною теорією нормального розподілу.

2.2 Багатовимірний нормальний розподіл

Система описується кількома змінними властивостей $(x_1, x_2, \dots, x_D)$, які співвідносяться між собою. Якщо хочемо змоделювати ці змінні разом як

одну модель Гаусса, потрібно використовувати багатоваріантну модель розподілу Гаусса [12]. ФЙЩ у БНР з розміром D визначається [9] як:

$$N(\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \exp\left[-\frac{1}{2}(x - \mu)^T(x - \mu)\right]$$

де D - номер розмірності, x - змінна, $\mu = E[x] \in R$, D - вектор середнього значення.

Використовуємо двовимірний нормальний розподіл (ДНР) як спрощений приклад для розуміння теорії БНР. Розподіл ДНР можна уявити у вигляді тривимірної кривої дзвоника з висотами, що представляють собою щільність ймовірності, показану на рисунку 2.6. Проекції тривимірної кривої дзвоника на площину x_1, x_2 є контурами еліпса, нанесеними на рисунку 2.7. Форма еліпсів показує кореляції між точками x_1 і x_2 , тобто наскільки одна змінна x_1 пов'язана з іншою змінною x_2 . $P(x_1, x_2)$ - це спільна ймовірність x_1 і x_2 . Для ДНР середній вектор μ є двовимірним вектором $\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, де μ_1 та μ_2 є незалежним середнім значенням x_1 і x_2 відповідно.

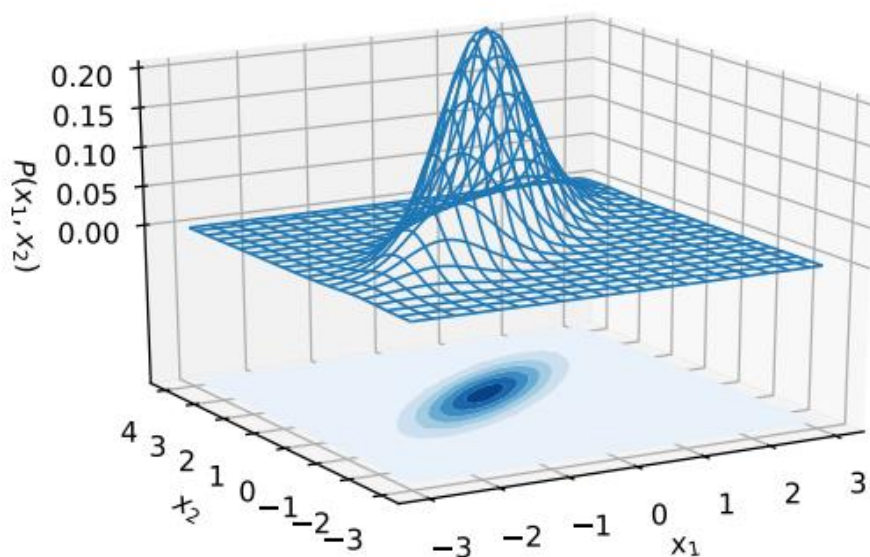


Рисунок 2.6 — Проекція тривимірної кривої представляє щільність ймовірності

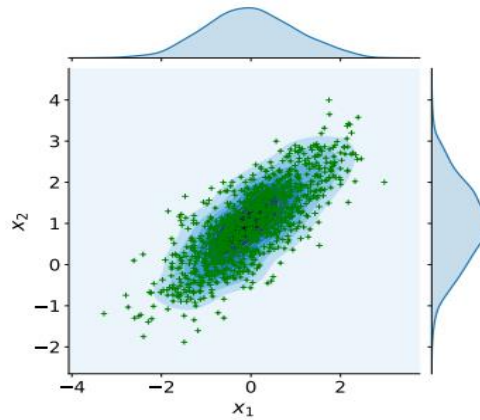


Рисунок 2.7 — Проекція тривимірної кривої на площину

Коваріаційною матрицею є $[\sigma_{00} \ \sigma_{01} \ \sigma_{10} \ \sigma_{11}]$, де діагональні члени σ_{00} і σ_{11} є незалежною дисперсією x_0 та x_1 відповідно. Внедіагональні σ_{01} та σ_{10} представляють кореляції між x_0 та x_1 [11]. ДНР виражається як:

$$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = N \left([\sigma_{00} \ \sigma_{01} \ \sigma_{10} \ \sigma_{11}] \begin{bmatrix} \mu_0 \\ \mu_1 \end{bmatrix} \right)$$

Інтуїтивно зрозуміло, що для задач регресії потрібна умовна ймовірність, а не спільна ймовірність. Якщо виріжемо зріз на тривимірній кривій, зображеному на рисунку 2.6, отримаємо умовний розподіл ймовірності $P(x_1 | x_2)$, показаний на рисунку 2.8. Умовний розподіл також є Гауссовим [11]

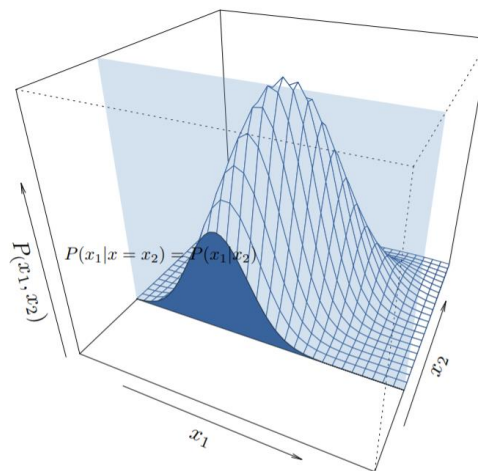


Рисунок 2.8 — Умовний розподіл ймовірності $P(x_1 | x_2)$

2.3 Функції ядра Гаусса

У регресії виходи функції повинні бути подібними, коли два входи знаходяться близько один до одного. Одним із можливих форматів рівняння є добуток векторів $A \cdot B = \|A\| \cdot \|B\| \cos \theta$, де θ – це кут між двома вхідними векторами [12]. Коли два вхідних вектора подібні, їх вихідне значення добутку є високим. Якщо функція визначена виключно через внутрішні добутки у вхідному просторі, то функція $k(x, x')$ є функцією ядра [11]. Найбільш широко використовувана ядра або коваріаційна функція - це квадратна експоненціальна функція (КЕФ) ядра. Ядро КЕФ де-факто є ядром за замовчуванням для гауссових процесів [3]. Це пояснюється тим, що його можна інтегрувати проти більшості функцій, які вам потрібні через його універсальної властивості. І кожна функція у своєму пріоритеті має нескінченно багато похідних [13]. Це також відоме як ядро радіальної базисної функції (РБФ) або функція ядра Гаусса, яка визначається як:

$$\text{cov}(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2}\right)$$

На рисунку 2.4 побудовано графік 20 незалежних гауссових векторів, з'єднавши 10 випадково вибраних точок вибірки кожного вектора по порядку за лініями. Замість того, щоб побудувати графік 20 незалежних, генеруємо 10 двадцяти варіативних нормальних розподілів з ідентичною коваріаційною функцією, показаною на рисунку 2.9.

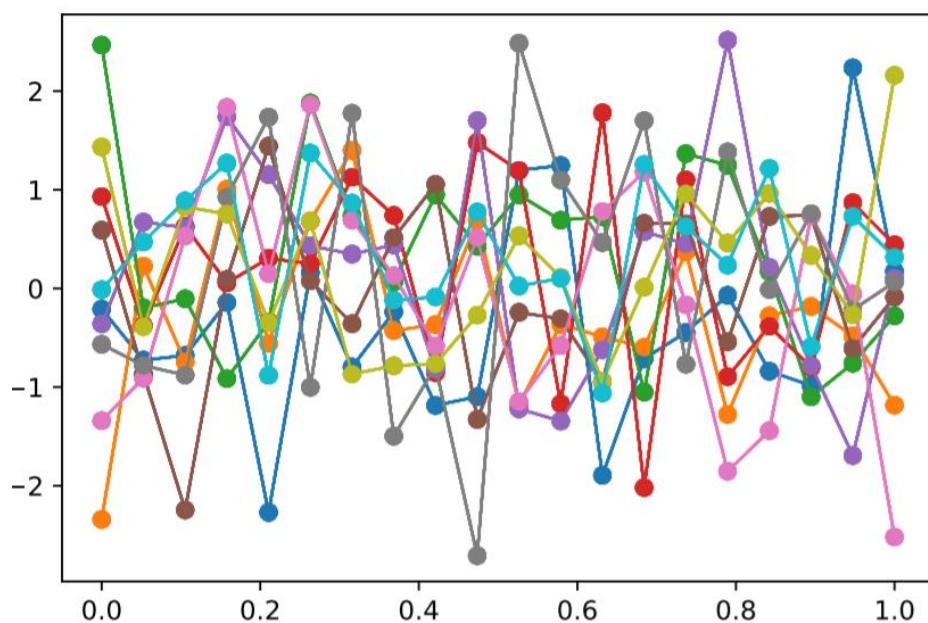


Рисунок 2.9 — Варіативні нормальні розподіли з коваріаційною функцією

Це те саме, що на рисунку 2.5, оскільки немає кореляції між точками, використовуючи ідентичність як функцію ядра [13]. Коли РБФ використовувався як коваріаційна функція, отримуємо гладкі лінії, показані на рисунку 2.10.

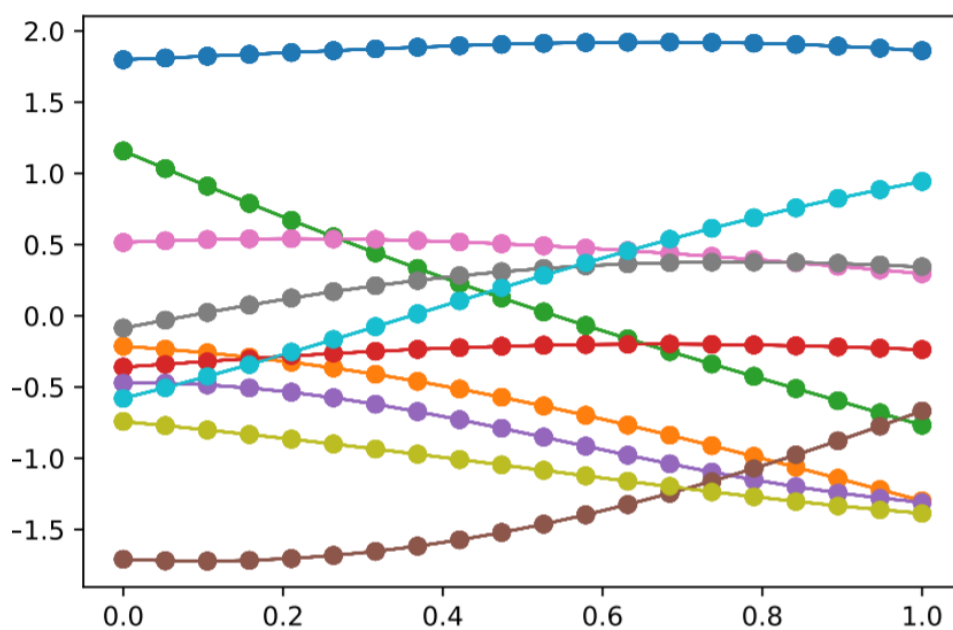


Рисунок 2.10 — Варіативні розподіли з радіальною базисною функцією

Додаючи функції коваріації, отримуємо більш плавні лінії, і вони починають виглядати як функції. Цілком природно розглянути можливість продовження збільшення розміру БНР. Тут розмірність відноситься до кількості мультиміжних БНР. Коли розмір БНР стане більшим, область інтересів заповнена більшою кількістю балів. Коли розмір стане нескінченним, з'явиться точка, яка представлятиме будь-яку можливу точку введення [14]. Використовуючи БНР з нескінченними розмірами, підганяємо функції з параметрами нескінченності для виконання завдань регресії. Тепер можна робити прогнози в будь-якому регіоні, що цікавить. На рисунку 2.11 будуюмо 200 зразків варіативного нормального розподілу, щоб отримати відчуття функцій з параметрами нескінченності. Це функції з попередніми ядрами, оскільки ще немає спостережуваних точок даних. Усі функції генеруються випадково моделлю БНР, що включає функції ядра як попереднє значення, перш ніж мати будь-які спостережувані точки даних.

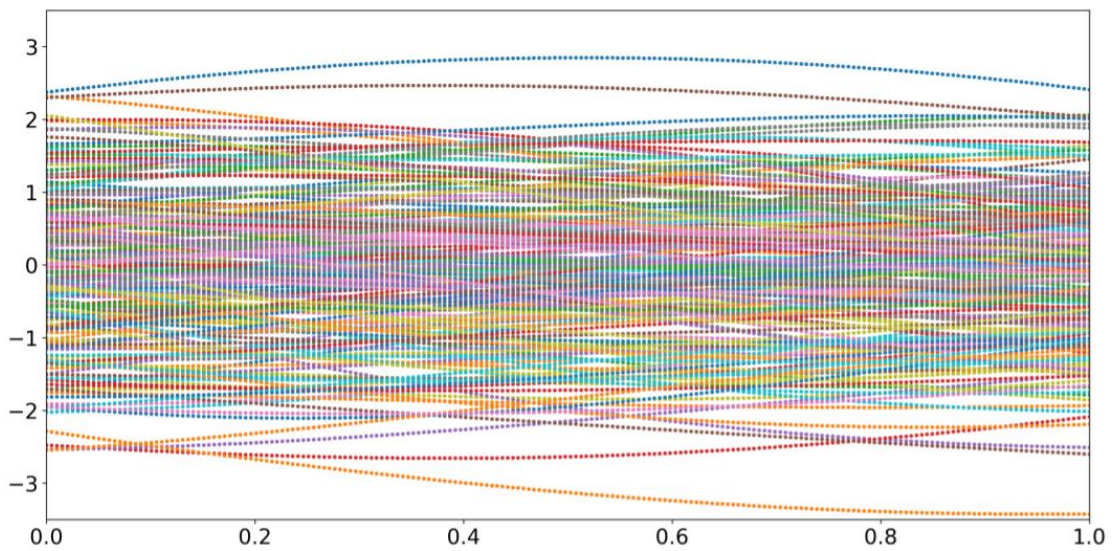


Рисунок 2.11 — Нормальний розподіл з параметром нескінченності

2.4 Непараметричні моделі

У цьому розділі пояснюється поняття непараметричної та параметричної моделі [10]. Параметричні моделі передбачають, що розподіл

даних можна моделювати з точки зору набору скінченної кількості параметрів. У регресії, враховуючи деякі точки даних, передбачити значення функції $y = f(x)$ з новим конкретним x . Якщо припустимо, що модель є лінійною регресією, $y = \theta_1 + \theta_2 x$, то потрібно знайти параметри θ_1 і θ_2 , щоб визначити функцію. У багатьох випадках припущення лінійної моделі не виконується, тому потрібна поліноміальна модель з більшою кількістю параметрів, наприклад $y = \theta_1 + \theta_2 x + \theta_3 x^2$. Використовуючи навчальний набір даних D з n спостережуваних точок, $D = [(x_i, y_i) \mid i = 1, \dots, n]$ для навчання моделі, тобто відображення x в y через базисні функції $f(x)$ [14]. Після процесу навчання припускаємо, що вся інформація даних фіксується параметрами функції, таким чином, що передбачення не залежать від навчального набору даних D . Тому, при проведенні регресій за допомогою параметричних моделей складність або гнучкість моделей обмежується кількістю параметрів. Якщо кількість параметрів моделі зростає разом із розміром спостережуваного набору даних, це непараметрична модель. Непараметрична модель не означає, що немає параметрів, а скоріше нескінченні параметри.

2.5 Процес Гаусса

У регресії є функція f , яку намагаємося змоделювати на основі спостережуваних точок даних D (набір даних для навчання) з невідомої функції f . Традиційні методи нелінійної регресії зазвичай дають одну функцію, яка найкраще відповідає набору даних. Однак може бути декілька функцій, які однаково добре відповідають спостережуваним точкам даних. Видно, що коли розмірність БНР була нескінченною, можна робити прогнози в будь-якій точці за допомогою цієї нескінченної кількості функцій. Ці функції є БНР, тому що це попереднє припущення. Формальніше кажучи, попередній розподіл цих нескінченних функцій - БНР. Попередній розподіл являє собою очікувані результати f над входами x без спостереження будь-яких даних. Замість нескінченної кількості функцій зберігаємо лише функції, які відповідають спостережуваним точкам даних. Коли з'являться нові спостереження,

використовуємо поточну функцію як попередню, а нові точки даних спостережень, щоб отримати нову функцію.

Гауссова модель процесів - це розподіл ймовірностей за можливими функціями, які відповідають набору точок. Ключові моменти процесів:

- функція оновлюється новими спостереженнями;
- гауссова модель процесу - це розподіл ймовірностей за можливими функціями, а будь - яка кінцева вибірка функцій спільно розподілена за Гауссом;
- середня функція, обчислена за розподілом можливих функцій є функція, яка використовується для прогнозування регресії.

Функція регресії, змодельована багатовимірним Гауссом, задається [15] як:

$$P(X) = N(\mu, K)$$

Без спостереження функція середнього значення за замовчуванням дорівнює $m(X) = 0$, враховуючи, що дані часто нормалізуються до нульового середнього. Модель процесів Гаусса є розподілом за функціями, форма (гладкість) яких визначається K . Якщо точки x_i і x_j вважаються подібними за ядром, вихідні дані двох точок, $f(x_i)$ і $f(x_j)$, очікується, що вони подібними. Процес проведення регресій за моделлю гауссівських процесів демонструється на рисунку 2.12. Враховуючи дані, що спостерігаються (червоні точки) та середню функцію f (синя лінія), оцінену цими точками спостережень, робимо прогнози в нових точках X як $f(X)$.

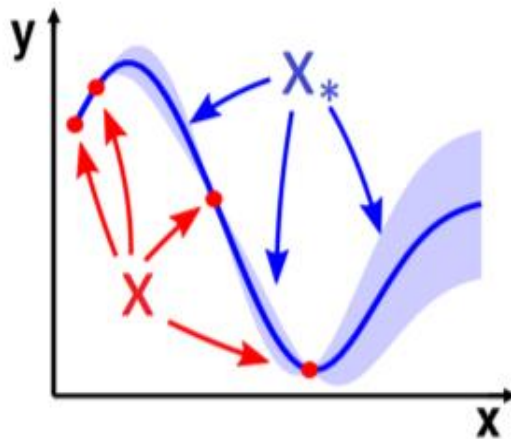


Рисунок 2.12 — Процес проведення регресій по гауссівським процесам

Спільний розподіл f і f_0 виражається як [15]:

$$\begin{bmatrix} f \\ f_0 \end{bmatrix} = N \left(\begin{bmatrix} m(X) \\ m(X_0) \end{bmatrix}, [K \ K_0 \ K_T \ K_{00}] \right)$$

де $K = K(X, X)$, $K_* = K(X, X_0)$, $K_{00} = K(X_0, X_0)$ і $(m(X), m(X_0)) = 0$.

Це спільне рівняння розподілу ймовірностей $P(f, f_0 | X, X_0)$ по f і f_0 , але регресії потребують лише умовного розподілу $P(f_0 | f, X, X_0)$ лише по f_0 . Виведення з спільного розподілу $P(f, f_0 | X, X_0)$ до умовного $P(f_0 | f, X, X_0)$ використовує теорему:

$$(f_0 | f, X, X_0) = N(K_T K_f K_{00} - K_T K^{-1} K_0)$$

У більш реалістичних ситуаціях немає доступу до справжніх значень функцій, але є їх шумні версії $y = f(x) + e$. Припускаючи, що існує адитивний незалежний і однаково розподілений гауссовий шум з дисперсією G_n^2 , пріоритетним для зашумлених спостережень стає $\text{cov}(y) = K + G_n^2 I$. Спільний розподіл спостережуваних значень і значень функції на нових випробувальних пунктах стає

$$\left[\frac{y}{f_0} \right] = N \left(0, \left[K + G_n^2 I \quad K_0 \quad K_T \quad K_{00} \right] \right)$$

Отримавши умовний розподіл, отримуємо прогнозовані рівняння для регресії гауссових процесів як [14]:

$$(f_0 | f, y, X_0) = N(f_0, cov(f_0))$$

де,

$$cov(f_0) = K_{00} - K_T [K + G_n^2 I]^{-1} K_0$$

У функції дисперсії $cov(f_0)$ можна відзначити, що дисперсія залежить не від спостережуваного виходу y , а лише від входів X і X_0 . Це властивість гауссового розподілу. Приклад результату був наведений на рисунку 2.13.

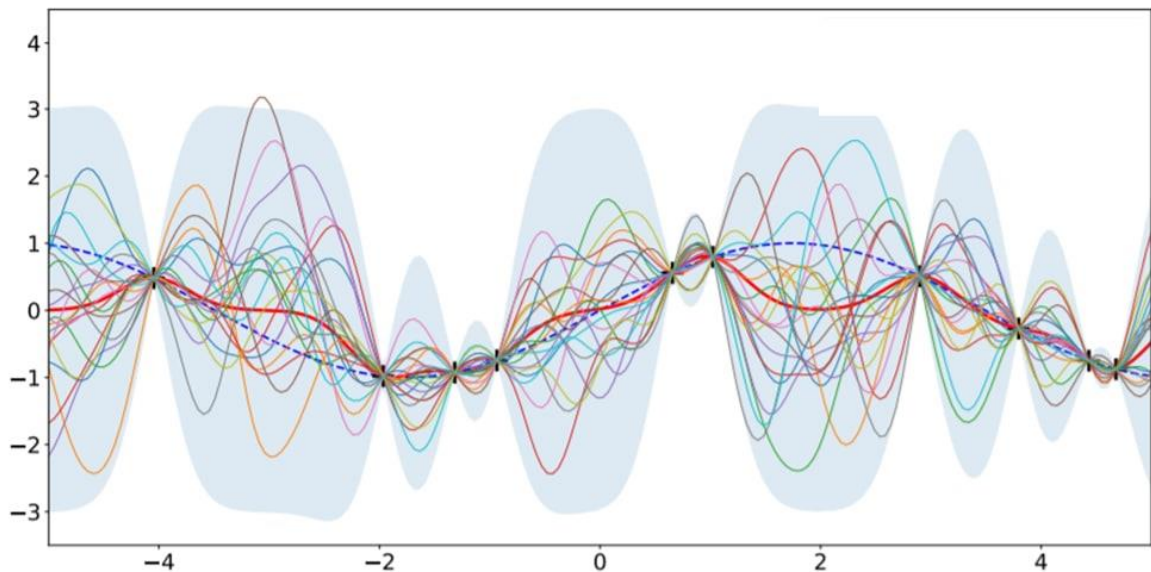


Рисунок 2.13 — Регресія рівномірного розподілу

Проведено регресію в межах $[-5, 5]$ області, що представляє інтерес. Спостережувані точки даних (навчальний набір даних) були сформовані з рівномірного розподілу між -5 і 5 . Будь-яка точка в межах даного інтервалу $[-5, 5]$ була рівною ймовірності бути витягнутою. Функції оцінювали в

рівномірно розташованих точках між мінус 5 і 5. Функція регресії складається із середніх значень, оцінених за допомогою моделі ГМП. Двадцять зразків середніх функцій також були нанесені в межах дисперсії.

2.6 Обробка новин

Завантажено всіх відповідних документів для кожної акції в наборі даних. Ці документи включають оголошення для преси, річні звіти, рекомендації аналітиків та загальні новини від понад 200 різних постачальників новин.

Множина, визначена в рівнянні складається з кожного окремого новинного документа (λ) для записів і містить час ($d(s,\lambda)$) і зміст ($C(s,\lambda)$) документа [19]. Дозволяємо ринковому часу реагувати на новини, ігноруючи будь-який документ, який відбувся протягом останніх Δt хвилин робочого дня (тобто $\text{time}(d(s,\lambda)) < \max(\text{time}()) - \Delta t$). Крім того, ігноруємо перші Δt хвилин робочого дня, оскільки очікуємо, що інвестори більше зосереджені на відкритті своїх позицій на день, а не на читанні останніх новин ($\min(\text{time}()) + \Delta t \leq \text{time}(d(s,\lambda))$) [21]:

$$A_{s,\Delta t} = \{A_1, A_2 \dots A_p\} A_{(s,\Delta t,\lambda)} = (d_{(s,\lambda)}, C_{(s,\lambda)})^{d_{(s,\lambda)}}$$

Усі документи попередньо обробляються для видалення чисел, URL-адрес, електронних адрес, безглузвих символів та форматування. Кожен термін у змісті $C(s,\lambda)$ документа має основу за допомогою алгоритму Стеммера Портера. Даний алгоритм видаляє суфікси зі слів, використовуючи суворі правила, які застосовуються до англійської мови, так що слова з однаковою основою вважаються одним і тим же словом. Наприклад, слова «finance», «financed», «finances» та «financing» однакові. Описаний алгоритм виконується, щоб зменшити кількість термінів, які необхідно досліджувати, і допомогти знайти подібні документи. Індекс базового терміну, визначений у формулі створюється з базовими термінами, які з'являються в документі

$(S(s,\lambda,\omega))$, і кількістю разів, коли вони з'являються в документі $(SC(s,\lambda,\omega))$, де ω - це вихідний ідентифікатор терміна[16]:

$$C_{s,\Delta\tau} = \{T_1, T_2 \dots T_p\} T_{(s,\Delta\tau,\lambda)} = (S_{(s,\lambda)}, SC_{(s,\lambda)})^{d_{(s,\lambda)}}$$

Для того, щоб класифікувати документи, спочатку необхідно класифікувати документи та визначити, які документи представляють більший інтерес. Побудова класифікатора, який передбачає, чи є документ цікавим, вимагає побудови навчальних і тестових наборів. Щоб переконатися, що ці документи в навчальному наборі мають щось спільне, необхідно проаналізувати терміни, що містяться в документах, і проранжувати терміни, які є найбільш цікавими. Згодом точність класифікаторів має бути перевірена шляхом порівняння прогнозів класифікаторів з фактичною категорією документа. Тому цей розділ розділений на підрозділи, що охоплюють категоризацію документів, навчальні та тестові набори, ранжування термінів, а також класифікацію та тестування.

Щоб визначити точність класифікатора, необхідно мати конкретні показники того, як ринок реагує на новини. Для цього необхідно провести аналіз часових рядів торгових даних і класифікувати кожен документ відповідно до того, як ринок поведився незабаром після його надходження. Часовий ряд повернення є одним із найцікавіших для інвесторів, оскільки він демонструє суму грошей, яку можна заробити. Однак на високих частотах неможливо передбачити прибуток, оскільки ринок занадто шумний. Реалізована мінливість, задана формулою $v_{(s,n,\rho,\Delta t)}$ частіше використовується у фінансовому співтоваристві для оцінки ризику володіння активом [20]:

$$v_{(s,n,\rho,\Delta t)} = \{v_1, v_2 \dots v_p\} v_{(s,\Delta\tau,\lambda)} = \left[\frac{1}{n} \sum_{j=0}^{n-1} R_{(s,n,\rho,\Delta t)}^p \right]^{\frac{1}{p}}$$

, де змінна n визначає кількість попередніх хвилин для підсумовування, а p є показником для повернення.

Параметри моделі оптимізовані з використанням торгових даних за попередній місяць, щоб гарантувати, що вони не відповідають умовам торгівлі даного місяця. Для календарного місяця січня використовуються параметри, оптимізовані з використанням усіх торгових даних по акції за календарний місяць грудень. Вибираються параметри, які формують функцію максимальної правдоподібності для даних даних. Це досягається шляхом максимізації логарифмічної функції правдоподібності, яка задана рівнянням [18]:

$$L(\beta) = -\frac{n}{2} \ln(2n) - \frac{1}{2} \sum_{j=1}^n (\ln(r_n))$$

для набору s , де є n записів у часовому ряді.

У даному випадку намагаємося оптимізувати модель після надходження новин. Тому застосовуємо, обчислюючи похибку прогнозу, задану рівнянням, яка є різницею між прогнозом та реалізованою волатильністю. Це підкреслює періоди, коли модель погано прогнозує волатильність. Аномальні помилки прогнозу з цими параметрами мають сильну кореляцію з надходженням новин про конкретні активи. Класифікуємо документи, частота яких корелює з аномальними помилками прогнозу, як цікаві. Для цього необхідно розрахувати середнє та стандартне відхилення похибки прогнозу за певний період. Змінна M в формулі визначає середню кількість торгових хвилин на місяць, використовуючи середню кількість торгових хвилин на робочий день для відповідної країни та помноживши на середню кількість торгових днів на місяць [17]:

$$M_{(s,n,\rho,\Delta t)} = \sqrt{\frac{\sum_{j=0}^{t-1} (\ln(\mu_n) - e_n)}{n}}$$

Параметри, які оцінюються в даному тестовому місяці, оптимізовані для максимізації функції логарифмічної правдоподібності для моделей під час навчального набору для кожного запасу. Навчальний набір складається з обмеженої кількості місяців, які відбулися до місяця тестування. Класифікатор, який використовується для класифікації документів протягом періоду навчання, навчається протягом періоду, який виключає місяці навчання та тестування. Це робиться для того, щоб класифікатор не використовував попередні знання, щоб визначити, як відреагує ринок після публікації новин. Параметри базової моделі оптимізовані протягом усього часового ряду в навчальному наборі для запасу [17]:

$$\sigma_{(s,n,\rho,\Delta t)} = \left(\sum_{k=1}^t (N_{(s,n,\rho,\Delta t)}^2) \right)$$

Параметри моделі оптимізуються протягом хвилин після випуску документа, класифікованого як кореляція з аномальною поведінкою ринку, використовуючи поріг у навчальному наборі:

$$\sigma_{(s,n,\rho,\Delta t)} = \sqrt{a_{s0} + \sum_{i=1}^p a_{si} (R_{(s,n,\rho,\Delta t)}^2) + \sum_{j=1}^q b_{sj} (R_{(s,n,\rho,\Delta t)}^2)}$$

Це пояснюється тим, що це єдиний випадок, коли модель дає прогноз, відмінний від базової моделі. У випадку, якщо не вдалося знайти параметри для покращення моделі порівняно з базовою моделлю, для запасу вибираються параметри за попередній місяць [16].

2.7 Оптимізація гіперпараметрів

На практиці моделі ГМП складніші. Функції ядра відіграють значну роль у ГМП. Вибір функцій ядра визначає практично всі властивості узагальнення моделі[4]. Існує багато функцій коваріації, які можна вибрати або зробити своїми руками для гауссового процесу залежно від конкретної проблеми. Ці критерії включають, чи є модель гладкою, чи вона розріджена, чи може вона різко змінюватися та чи потрібно її диференціювати [5]. Більш детальну інформацію щодо вибору функції ядра/коваріації для гауссового процесу можна знайти в [5]. У ядрах оптимізація гіперпараметрів є важливою. Тут використовується найбільш широко використовуване ядро, РБФ, як приклад для пояснення оптимізації гіперпараметрів. Загальна функція РБФ визначається як:

$$K(x_i, x_j) = G_f^2 \exp\left(-\frac{1}{2}(x_i - x_j)^T (x_i - x_j)\right)$$

де G_f^2 і l — гіперпараметри [8]. Вертикальна шкала G_f^2 описує, на скільки функція може охоплювати значення. Горизонтальна шкала l показує, як швидко зменшується кореляційний зв'язок між двома точками при збільшенні їх відстані. Вплив l був показаний на рисунках 2.14-2.16. Вищий l забезпечував більш гладку функцію, а менший l давав функцію Вігльєра.

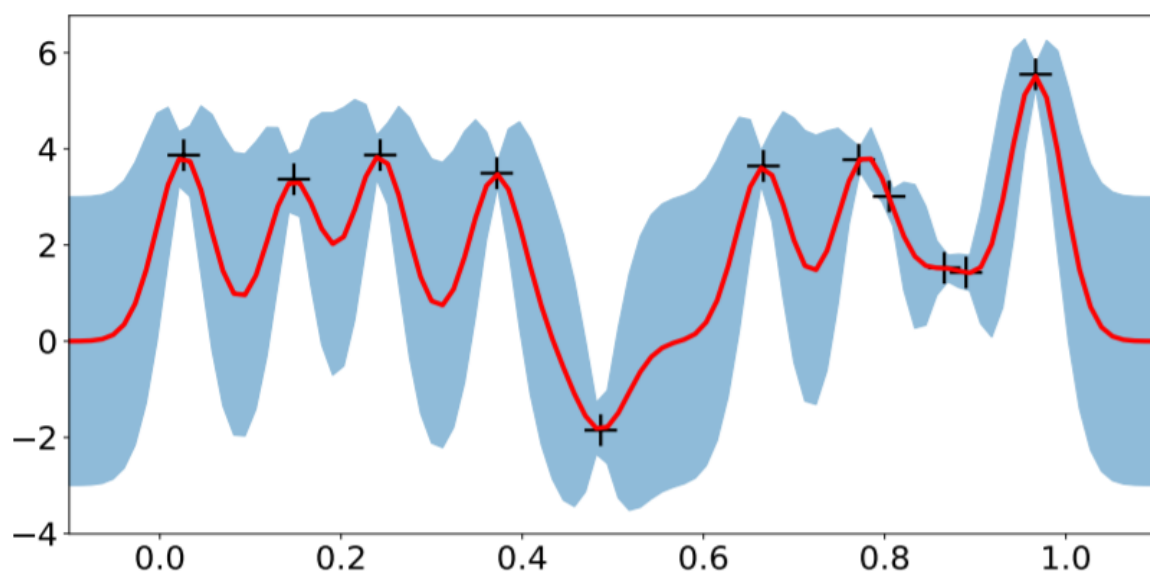


Рисунок 2.14 — Функція Вігльєра при зменшеній відстані

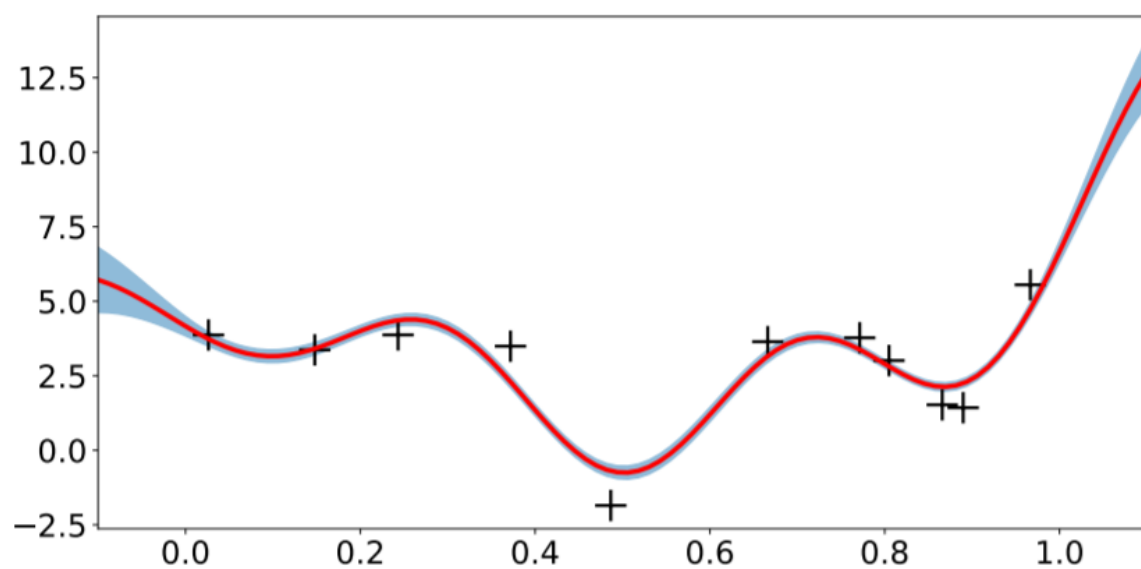


Рисунок 2.15 — Вплив рівномірних гіперпараметрів

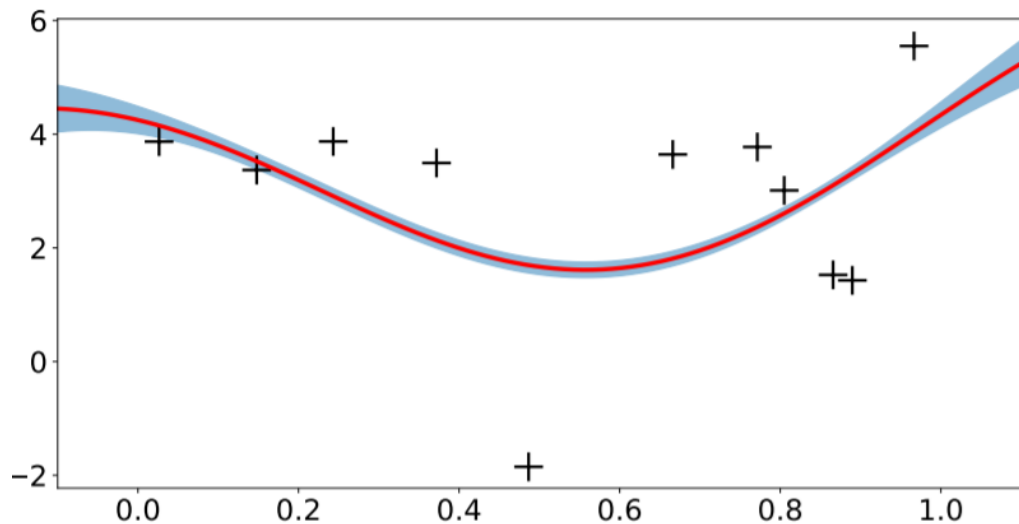


Рисунок 2.16 — Кореляційний зв'язок збільшенні їх відстані

Оптимізовані гіперпараметри Θ визначаються[11] як:

$$\theta = \arg (\max (\log p(x, \theta)))$$

Таким чином, розглядаючи гіперпараметри, є більш загальним рівнянням прогнозів у нових точках тестування [5]

$$(f_* | X, y, X_*, \Theta) = N(f_*, \text{cov}(f_*))$$

Після вивчення та налаштування гіперпараметрів прогнозна дисперсія $\text{cov}(f_*)$ залежить не тільки від вхідних даних X і X_* , але й від виходів y [1]. За допомогою оптимізованих гіперпараметрів $G_f = 0,0067$ та $l = 0,0697$, результат регресії спостережуваних точок даних на рисунку 2.13 був показаний на рисунку 2.17

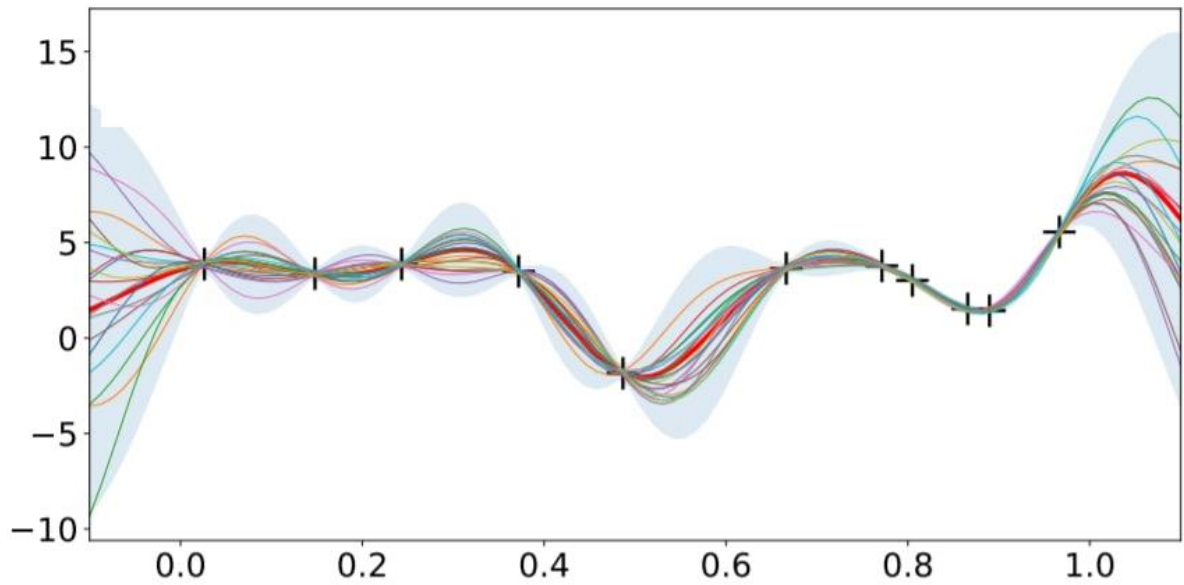


Рисунок 2.17 — Регресія з оптимізованими гіперпараметрами

Загальна схема запропонованого математичного рішення зображена на
рисунку 2.18

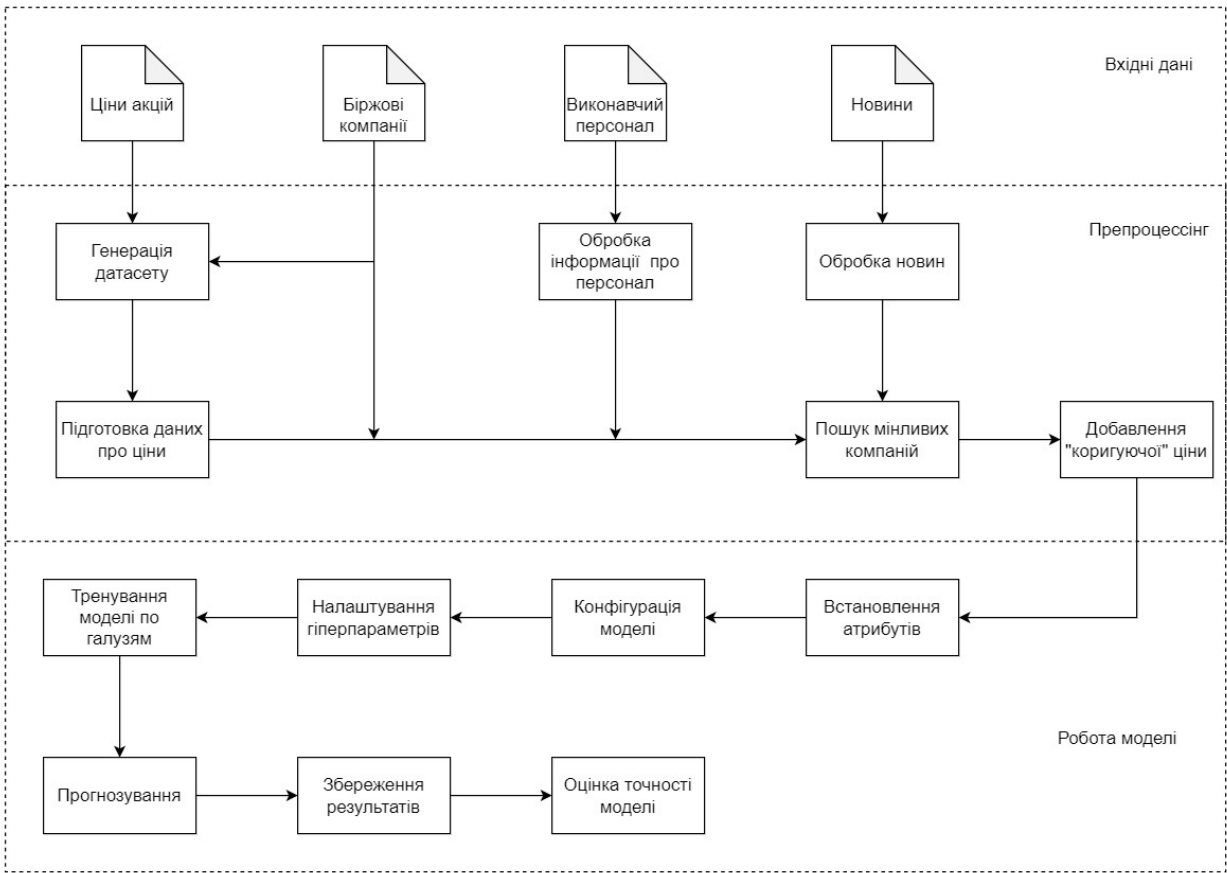


Рисунок 2.18 — Загальна схема математичного рішення

2.8 Висновки до розділу

У другому розділі було описано обрана модель, яка використовується для вирішення задач регресії та класифікації. Продемонстровано математичне забезпечення для обробки новин. Було проаналізовано та налаштовано гіперпараметри для оптимізації алгоритму.

3 СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз програмного забезпечення

Програмне забезпечення представляє собою веб-додаток (рисунок 3.1). Для реалізації даного програмного забезпечення, використано клієнт-серверну архітектуру, адже такий підхід вважається найоптимальнішим у сфері веб-розробки (детальніша архітектура програмного забезпечення представлена у додатку В).



Рисунок 3.1 — Клієнт-серверна архітектура додатку

Клієнт-серверна архітектура включає в собі декілька частин:

- клієнтська – ця частина призначена для відображення даних. Для її написання було використано JavaScript — мова програмування для створення інтерактивних Web-сторінок. Вона має динамічну типізацію мови. Також надає зручний для користування та розробки програмний інтерфейс, з доступом до DOM дерева сторінки;
- серверна – головна частина програмного забезпечення, яке включає в собі математичне забезпечення та бізнес логіку. Для реалізації даної частини було взято Java – це кросплатформенна строго типізована об'єктно-орієнтована мова програмування;
- база даних – це частина, яка відповідає за збереження, зміну або видалення даних у організованих системах. Так як PostgreSQL є найпопулярнішою системою управління базами даних, то вона була обрана

для цієї частини. Дана СУБД є відкритою системою, яка представлена у вигляді реляційної бази даних;

– фреймворк алгоритмів машинного навчання для класифікації та регресії – Weka. За допомоги даного набору інструментів для машинного навчання можливо на основі часових рядів прогнозувати ціни на певний торговий день для компаній, які були визначені раніше класифікаційною моделлю.

3.2 Архітектура програмного забезпечення

3.2.1 Головні класи програмного забезпечення

Controller – це зв’язковий компонент архітектури MVC між клієнтською та серверною частиною. Він керує додатком, отримуючи запит від користувача, а потім перетворює ці запити у команди та передає їх серверній частині, де виконується математичне забезпечення.

Присутні такі контролери, сервіси, репозиторії в даному програмному забезпеченні (діаграма класів та схема бази даних відображені у додатку А і Б відповідно):

- AutomatedExecutionController – контролер для отримання даних про категорії, індекси, сектори, табличну та графічну інформації для компаній;
- AutomaticUpdateController – контролер для автоматичного оновлення з інтервалами: хвилина, година, день, тиждень, місяць та рік;
- CreateFileController – контролер для створення файлу з датасетами та звітами;
- GraphController – контролер для графічного відображення даних моделей;
- ModelController – контролер для роботи з класифікаційними, регресійними та комбінованими моделями;

- ProfitController – контролер для збору та відображення статистичних даних по відпрацюванні моделей та загальною інформації по доступним даним для кожної компанії, секторів та індексів;
- ShowViewController – це контролер для взаємодії серверної та клієнтської частини;
- UpdateController – це контролер для збору та оновлення даних для компанії, секторів, індексів, цін, та звітів;
- ClassifierFuturePredictionPrice – це сервіс для класифікації тікерів компаній, які будуть включені у датасет для регресійної моделі, результатом якої будуть майбутні прогнозовані ціни;
- ClassifierPastPredictionPrice – це сервіс для класифікації тікерів компаній, які будуть включені у датасет для регресійної моделі, результатом якої будуть минулі прогнозовані ціни;
- PricePredictedChange – це сервіс для математичного обчислення зміни ціни для певної компанії протягом п'яти останніх робочих днів біржі.
- ModelEvaluationMapper – це сервіс для перетворення результатів класифікаційної моделі;
- DataForGraphService – сервіс для структуризації даних, що будуть графічно відображенні на клієнтській частині;
- RegressionPredictor – це сервіс для прогнозування цін акцій компаній використовуючи регресійну модель. Даний сервіс може відтворюватися самостійно, а також після класифікаційної моделі у разі відтворення комбінаційної моделі;
- Forecaster – це сервіс для прогнозування та порівняння цін за допомоги регресійної моделі;
- FileWorker – це допоміжний сервіс для збереження результатів регресійної моделі у вигляді файлу.

3.2.2 Кінцеві точки зв'язку

Кінцеві точки зв'язку (endpoints) – це інтерфейс для взаємодії клієнтською частини з серверною. Кожен endpoint повинен мати URL-адресу для ідентифікації ресурсу. За допомогою них можна отримати ресурси, які передаються з серверної частини до клієнтської. Найголовніші кінцеві точки будуть описані в таблиці 3.1

Таблиця 3.1 — Кінцеві точки додатку

Endpoint і його контролер	Опис
/getCategories AutomatedExecutionController	Отримання категорій, індексів, секторів компаній для відображення діаграми результатів прогнозування моделі.
/getTableCompaniesInfo AutomatedExecutionController	Отримання даних компаній для відображення у таблиці прогнозованих компаній: – тікер. – назва компанії; – вчорашня ціна; – ціна на даний момент; – три прогнозовані ціни на наступні дні. На вхід подається вибрана категорія користувачем, для відображення таблиці з прогнозованими компаніями.
/getGraphicCompanyInfo AutomatedExecutionController	Отримання усіх історичних і прогнозованих цін компаній, яка була обрана користувачем. На вхід подається тікер компанії, історичні та прогнозовані ціни, які будуть графічно відображатися

Продовження таблиці 3.1

Endpoint і його контролер	Опис
/autoHourUpdate AutomaticUpdateController	Автоматичне щогодинне оновлення за допомоги schedule, для ініціалізації або оновлення даних: – живих цін (останніх доступних цін компаній, що котируються на міжнародних біржах Nasdaq та NYSE); – останніх новин достовірних джерел для відповідних компаній.
/autoDailyUpdate AutomaticUpdateController	Автоматичне щоденне оновлення за допомоги schedule, для ініціалізації або оновлення даних: – цінних паперів; – обчислення статистичних даних про кожную компанію; – запуск класифікаційної, регресійної та комбінованої моделей; – зчитування новин; – оновлення списку достовірних джерел новин та статей.
/autoMonthlyUpdate AutomaticUpdateController	Автоматичне щоденне оновлення за допомоги schedule, для ініціалізації або оновлення даних: – список доступних компаній; – інформація про компанію; – список S&P компаній; – список новинних джерел.

Продовження таблиці 3.1

Endpoint і його контролер	Опис
manualExecuteCombineModel ModelController	Виконання комбінованої моделі, яке включає в собі спершу виконання класифікаційної моделі, результат якої передають параметром для відтворення регресійної моделі.
manualExecuteTickersModel ModelController	Виконання комбінованої моделі для заданих компаній однієї із доступних для користувача моделей. На вхід подається список тікерів компаній, які будуть спрогнозовані.
runModel ModelController	Виконання всіх моделей (класифікаційної, регресійної та комбінованої), які будуть відбуватися у паралельному режимі.
manualExecuteCategoryModel ModelController	Виконання комбінованої моделі для заданої категорії. На вхід подається категорія, компанії якої будуть спрогнозовані
periodExecuteCategoriesModel ModelController	Виконання комбінованої моделі для заданого періоду. На вхід подається період для прогнозування цін компаній.
periodClassifyExecuteCombine ModelController	Виконання класифікаційної моделі для заданого періоду. На вхід подається період для прогнозування цін компаній.
/saveLastModelResults ModelController	Збереження останніх результатів виконання моделі до файлу.
/updateProfit ProfitController	Обчислення статистики коректності відпрацювання усіх моделей.

Продовження таблиці 3.1

Endpoint і його контролер	Опис
/filterModelCompanies ProfitController	Оптимізація компаній, які не мають необхідну кількість даних для прогнозування цін моделями.
/getCategoriesProfit ProfitController	Обчислення статистики коректності відпрацювання усіх моделей для певної категорії. На вхід подається категорія для обчислення статистики моделей.
/updateNyseList UpdateController	Мануальна ініціалізація та оновлення списку компанії, які котируються на NYSE біржі.
/updateCompanyInfo UpdateController	Мануальна ініціалізація та оновлення інформації для усіх компаній.
/updateSecuritiesInfo UpdateController	Мануальна ініціалізація та оновлення інформації для цінних паперів компаній.
/updateSecuritiesPrice UpdateController	Мануальна ініціалізація та оновлення цін для цінних паперів компаній.
/updateIndexList UpdateController	Мануальна ініціалізація та оновлення списку усіх компаній.
/getUpdateNews UpdateController	Мануальна ініціалізація та оновлення усіх новин достовірних джерел.
/getLastNews UpdateController	Мануальна ініціалізація та оновлення останніх доступних новин достовірних джерел.
/updateEffectiveDateNews UpdateController	Мануальна ініціалізація оновлення дати для зчитування останніх новин.
/updateLivePrices UpdateController	Мануальна ініціалізація та оновлення живих цін (останніх цін на даний момент).
/getFileByName UpdateController	Отримання файлу з результатами виконання моделі. На вхід подається ім'я файлу.

Продовження таблиці 3.1

Endpoint і його контролер	Опис
/updateCompanyDataQualities UpdateController	Мануальна ініціалізація та оновлення статистики про наявність та коректність даних компаній для прогнозування.
/updateIndexes UpdateController	Мануальна ініціалізація та оновлення індексів компаній, які котируються на американських Nasdaq та NYSE біржах.
/updateNasdaqExchange UpdateController	Мануальна ініціалізація та оновлення списку компанії, які котируються на Nasdaq біржі.
/showCharts ShowViewController	Графічне відображення історичних та прогнозованих цін акцій компанії. На вхід подається ім'я файлу.
/showMainPage ShowViewController	Відображення стартової сторінки для серверної частини, яка застосовується для мануального запуску моделей.
/dataset ShowViewController	Відображення усіх доступних датасетів класифікаційної, регресійної та комбінованої для прогнозування цін акцій .
/models ShowViewController	Відображення результатів виконання всіх моделей (комбінованої, класифікаційної та регресійної) з прогнозованими цінами акцій.
/model ShowViewController	Відображення результатів виконання конкретної моделі з прогнозованими цінами акцій. На вхід подається тип моделі.
/viewFile ShowViewController	Відображення файлу з результатом виконання моделі та її детальної статистики. На вхід подається ім'я файлу.

Продовження таблиці 3.1

Endpoint і його контролер	Опис
/downloadFile ShowViewController	Збереження файлу у форматі csv з результатом виконання моделі до системного сховища. На вхід подається ім'я файлу.
/qualityReport ShowViewController	Відображення обчислень статистики коректності відпрацювання усіх моделей та наявності валідних даних компаній для прогнозування.
/selectTickers ShowViewController	Відображення усіх доступних тікерів для вибору компаній, які будуть виконані методом manualExecuteTickersModel.

3.3 Аналіз безпеки даних користувача

Програмне забезпечення використовує дані для прогнозування акцій компаній, які не є загальнодоступними для кожного в Інтернеті. Тому було забезпечено надійність додатку, додавши за допомоги модулю Spring Security авторизацію та аутентифікацію користувача.

Для безпечного з'єднання з клієнтською та серверною частинами використано популярний HTTPS протокол. Він надає гарантії, що інформацію буде безпечно передано, адже має допоміжне шифрування між протоколами TCP та HTTPS, що робить його криптографічно стійким.

3.4 Інструкція для користувача та демонстрація проекту

У рамках магістерської дисертації було розроблено інтелектуальну систему прогнозування біржових цін акцій компаній на основі аналізу часових рядів коливань курсу з зручним графічним інтерфейсом, тому для кращої наглядності добавлено скриншоти з демонстрацією даного додатку додано до інструкції користувача.

Перейшовши по посиланню, користувач знаходиться на головній сторінці застосунку (рисунок 3.2).

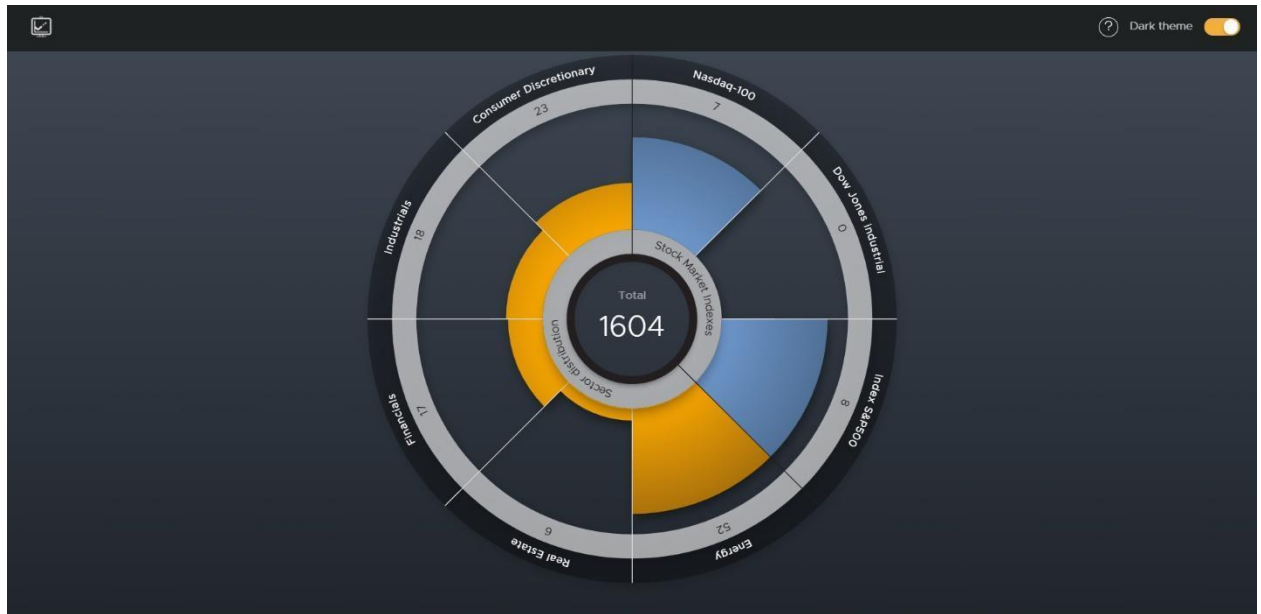


Рисунок 3.2 — Головна сторінка застосунку

Відображена гістограма, яка розділена на вісім секторів для двох підгруп. Перша підгрупа – це фондові індекси (позначені синім кольором):

- «Dow Jones industrial»;
- «Nasdaq-100»;
- «Index S&P500».

Друга підгрупа – це галузі, які мають найбільшу кількість найвпливовіших американських компаній (позначені помаранчевим кольором):

- «Energy» (енергетична галузь);
- «Real Estate» (галузь нерухомості);
- «Financials» (фінансова галузь);
- «Industrials» (індустріальна галузь);
- «Consumer Discretionary» (споживча галузь).

На головній сторінці користувач може обрати сектор з прогнозованими компаніями, вибрати кольорову тему додатку, перейти на форму з детальною інформацією (рисунок 3.3) та переглянути статистику по загальній кількості доступних компаній для прогнозування.

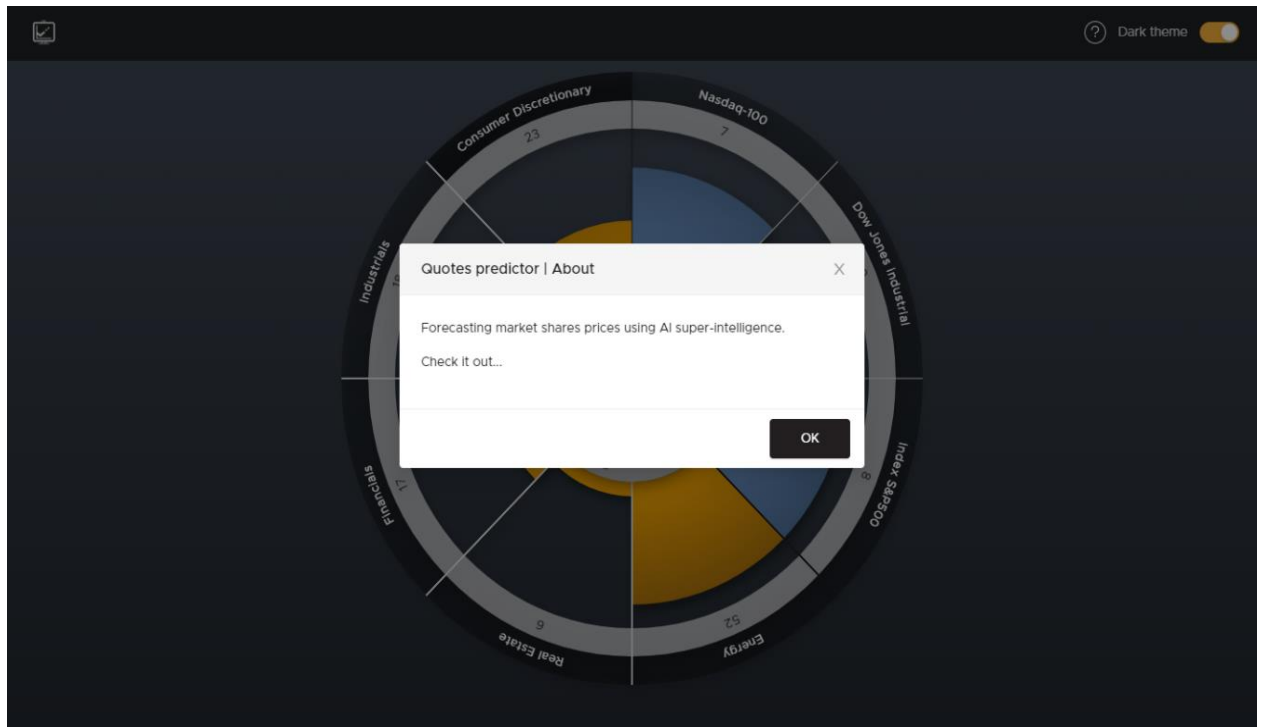


Рисунок 3.3 — Форма з детальною інформацією

Під назвою кожного сектору відображається кількість компаній, які змінять ціну протягом трьох днів, що було спрогнозовано моделлю, що відтворюється автоматично щоденно. На основі цих даних заповнюються індикатори гістограми по підгрупах. Чим більше прогнозованих компаній в порівнянні з іншими секторами у підгрупі тим більший індикатор. При обранні сектору відкривається таблиця з прогнозованими компаніями сектору з детальною інформацією (рисунок 3.4).

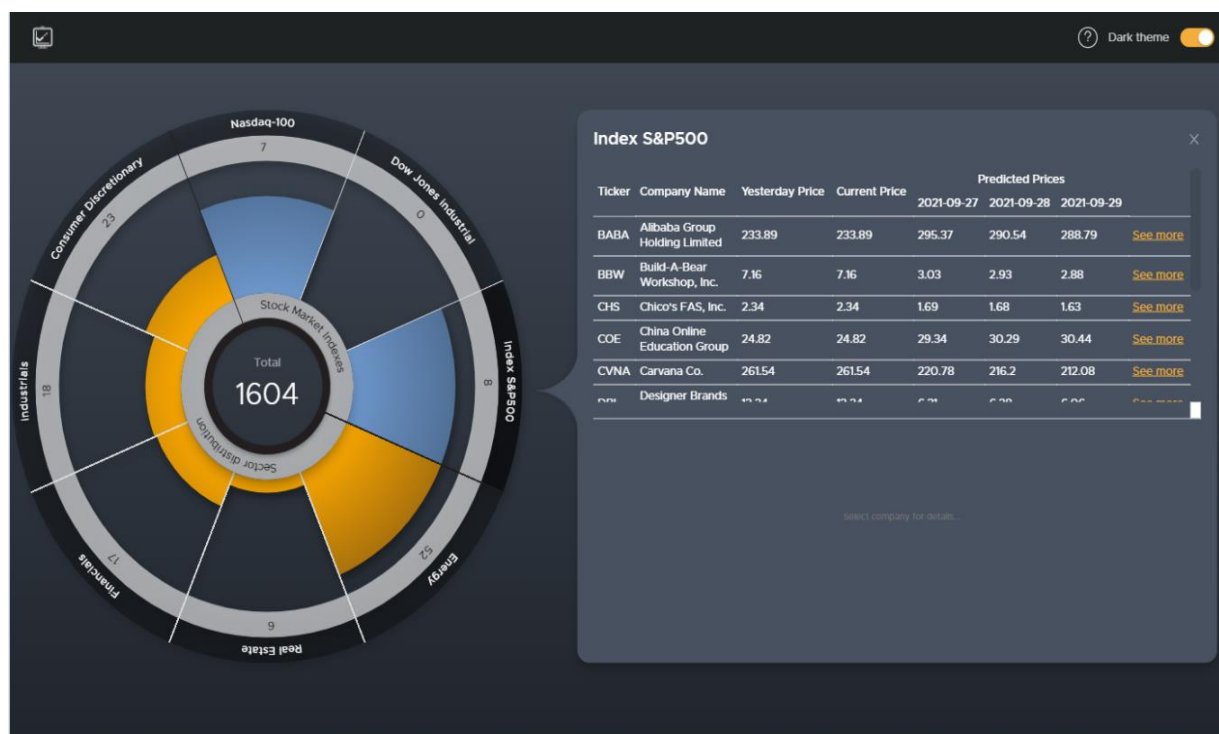


Рисунок 3.4 — Таблиця з прогнозованими компаніями сектору

Для графічного відображення історичних та прогнозованих цін акцій конкретної компанії на графіку натисніть «See more» (рисунок 3.5) або скористуйтесь підказкою, яка продемонстрована у вільній частині блоку під таблицею.

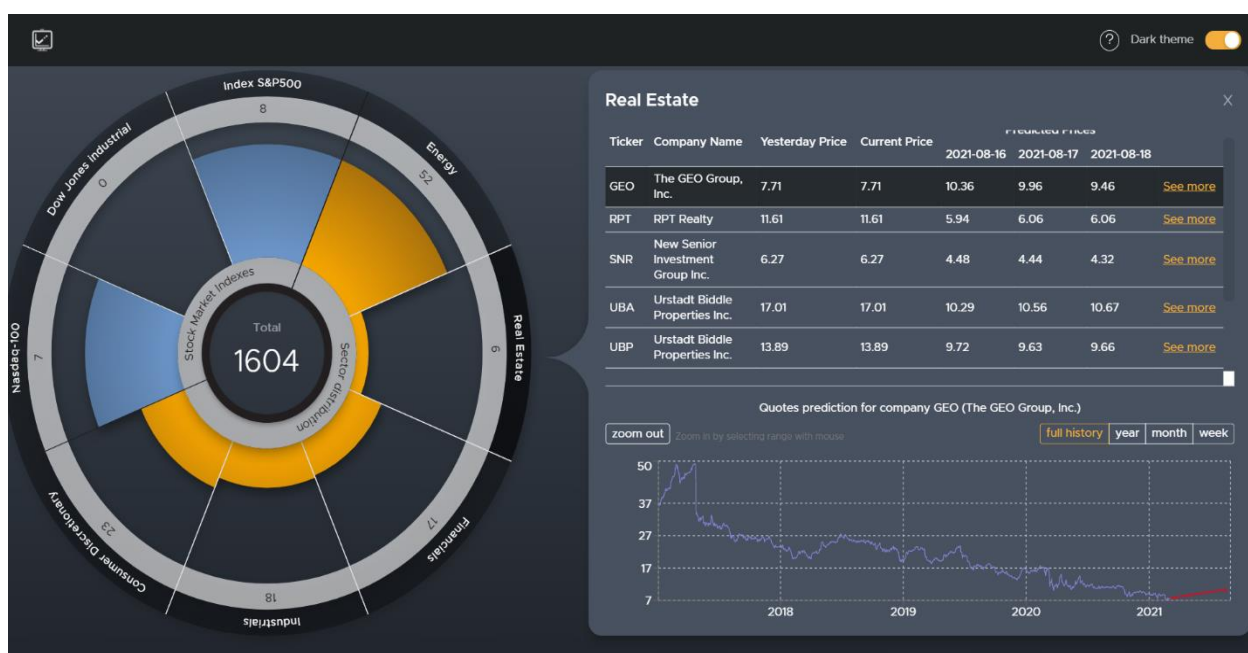


Рисунок 3.5 — Таблиця з графіком цін акцій компанії

Графік підтримує декілька режимів показу даних:

- тижневе відображення історичних та прогнозованих цін акцій обраної компанії (рисунки 3.6);

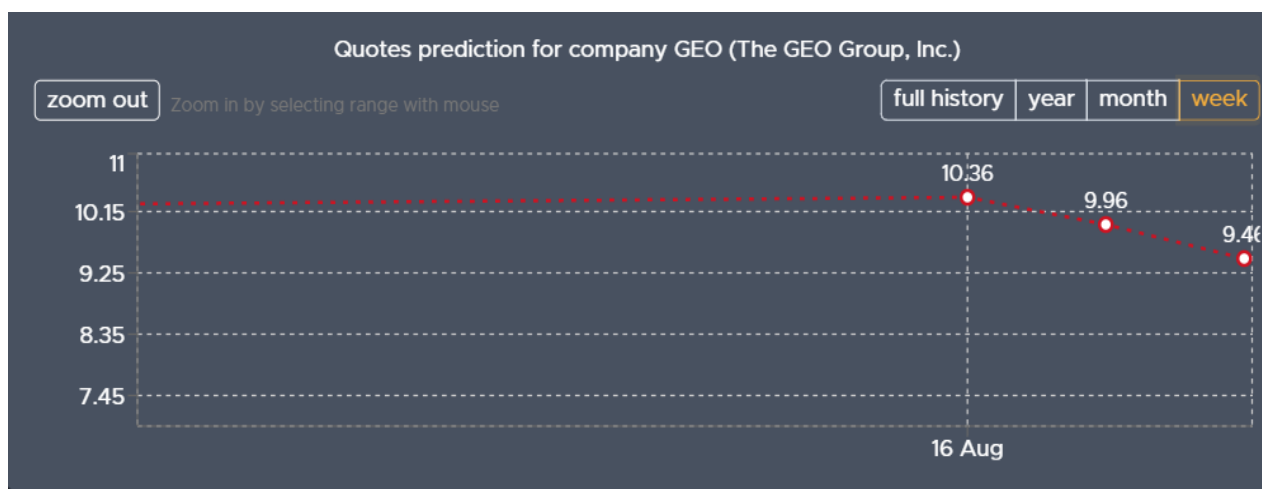


Рисунок 3.6 — Тижневе відображення результатів

- місячний графік цін акцій The Geo Group, Inc (рисунки 3.7);

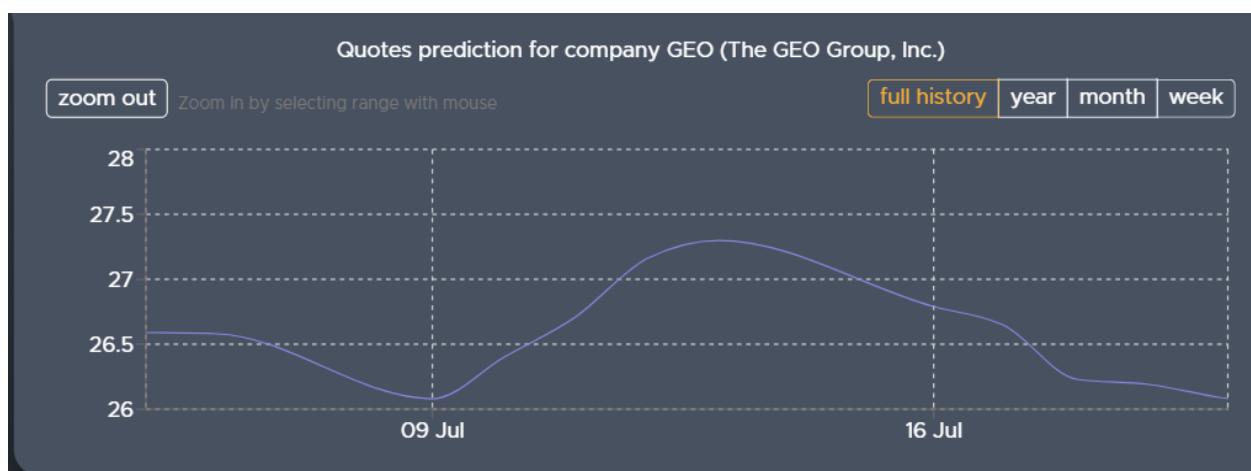


Рисунок 3.7 — Місячне відображення результатів

- річне відображення історичних та прогнозованих цін акцій обраної компанії (рисунки 3.8);

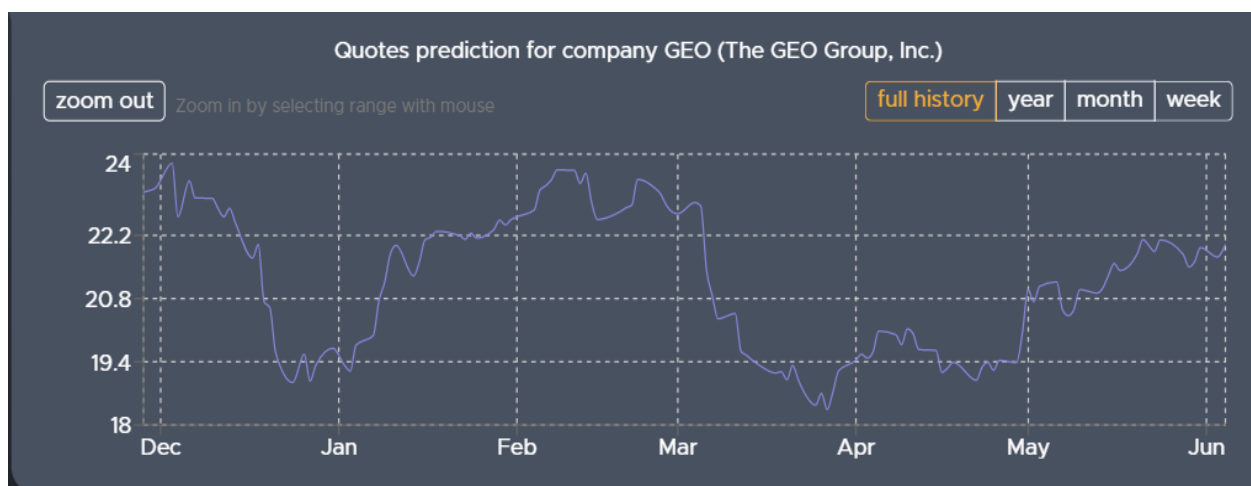


Рисунок 3.8 — Річне відображення результатів

– графік довольного періоду, який налаштовується користувачем курсором мишки або іншим мультимедійним пристроєм (рисунок 3.9);

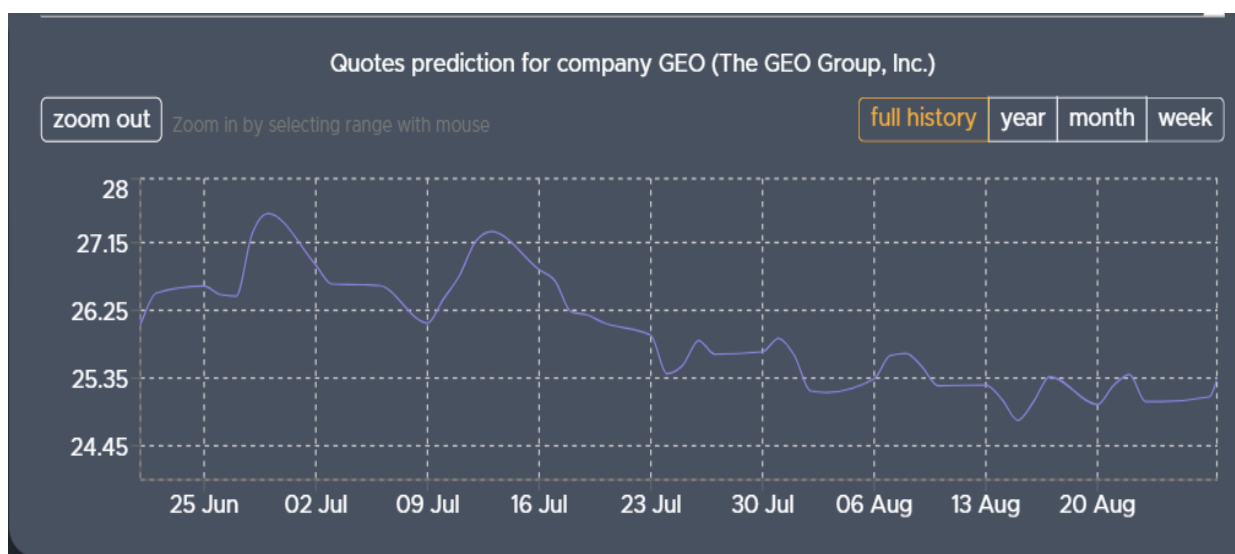


Рисунок 3.9 — Відображення результатів довільного періоду

– графік довільного періоду, який налаштовується користувачем курсором мишки або іншим мультимедійним пристроєм (рисунок 3.10).



Рисунок 3.10 — Графік результатів повного періоду

3.5 Генерація датасету

Генерація набору даних забезпечується певною кінцевою точкою, яка вимагає від користувача надсилати параметри запиту у форматі JSON. Нижче наведено приклад такого запиту у форматі JSON (рисунок 3.11).

```
{
  "timePeriodSplitter": "timeSplitter",
  "startDate": "2017-02-28",
  "endPeriod": "2020-03-29",
  "splitFactor": "DAY",
  "splitCount": 1,
  "companySelector": "selectedCompanies",
  "dataProviders": [
    "dataProviderPriceClosed"
  ],
  "dataSetType": "COMPANY"
}
```

Рисунок 3.11 — Запит для генерації датасету у форматі JSON

StartDate – дата початку ціноутворення для компанії у форматі «РРРР-ММ-ДД» (у середньому кожна компанія має від трьох торгових років).

EndPeriod – дата закінчення ціноутворення для компанії у форматі «РРРР-ММ-ДД» (дані доступні до першого числа поточного місяця).

Різновиди розбиття цін на кластери (splitFactor):

- «DAY» – вибір ціни для кожного вказаного дня в (startDate-endPeriod).
- «MONTH» – вибір ціни для кожного вказаного місяця.
- «YEAR» – вибір ціни для кожного вказаного року в (startDate-endPeriod).

SplitCount – період часу, через який потрібно взяти дані. Можливий діапазон даного значення від одиниці до максимальної кількості доступних торгових днів компанії. Наприклад, якщо поставлено два, то ціна чергується через кожні 2 дні.

TimePeriodSplitter - параметр за замовчуванням, який генерує список дат на основі цих параметрів (startDate, endPeriod, splitFactor, splitCount).

Доступні кластери компаній (companySelector):

- selectedCompanies – вибір певних, раніше визначених, 10 компаній, які мають найбільшу кількість торгових днів;
- AllCompanies – всі доступні компанії, які зберігаються у базі даних;
- AllIndexCompanies – компанії S&P500 та Nasdaq;
- AllNotIndexCompanies – обрано усі компанії, які не входять до вищезазначених бірж.

DataProviders – це характеристика, яка формує ціни по певним критеріям. Може мати кілька параметрів, якщо для dataSetType встановлено значення COMPANY:

- dataProviderPriceClosed – вибір даних за ціною (закриття) на момент закриття біржі;
- dataProviderNews – аналогічно до першого, але з новинами;
- dataProviderIndustry – аналогічно з розділенням по галузям;
- dataProviderIndustryPriceMedian – ціноутворення по індустріям, де нулі замінюються медіанами.

Типи генерацій датасету та збереження їх (dataSetType) :

- «COMPANY» – окремий файл результатів для кожної компанії;
- «CUSTOM» – розбиття компаній по галузям та індексам;

- «PERIOD» – формування файлів по визначених періодах.

3.6 Експериментальні дослідження

Для побудування моделей були використані такі фінансові дані, як:

- основна інформація про кожну компанію (назва компанії, номер, адреса, місцевість, галузь, номер телефону тощо);
- інформація про виконавчий персонал (посада, ім'я, вік, стать, рік затвердження, тощо);
- основна фінансова інформація для кожної акції, як волатильність, ціна відкриття (open price), ціна закриття (close price), максимальне значення ціни акції протягом дня (high price), мінімальне значення ціни акції протягом дня (low price);
- новини для кожної компанії (новини включають короткий опис, зображення, текст, посилання на джерело новин, тощо);
- інформація про фінансові звіти, як форма 10-K/Q (включаючи основну частину числових даних зі звіту, як-от різні типи активів та зобов'язань).

Для демонстрування прогнозування моделі було взято кластер компаній *selectedCompanies* – це набір певних, раніше визначених, 5 компаній, які мають найбільшу кількість торгових днів у базі даних:

- «IBM» (International Business Machines);
- «A» (Agilent Technologies);
- «AAP» (Advance Auto Parts);
- «KEY» (KeyCorp);
- «UPS» (United Parcel Service).

Щоб перевірити ефективність алгоритму протягом довгого періоду модель прогнозувала 250 значень (що відповідає календарному торговому року) з горизонтом у три дні та збереженням проміжкових результатів. Далі наведено графічне порівняння прогнозованих даних з реальними цінами.



Рисунок 3.12 — Графік реальних і прогнозованих цін «IBM»
(International Business Machines)

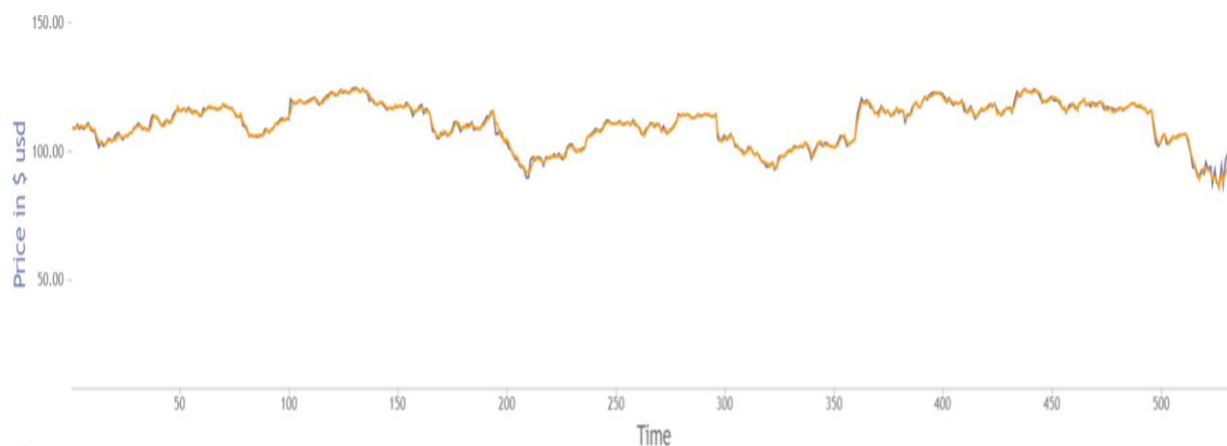


Рисунок 3.13 — Графік реальних і прогнозованих цін «UPS» (United
Parcel Service)

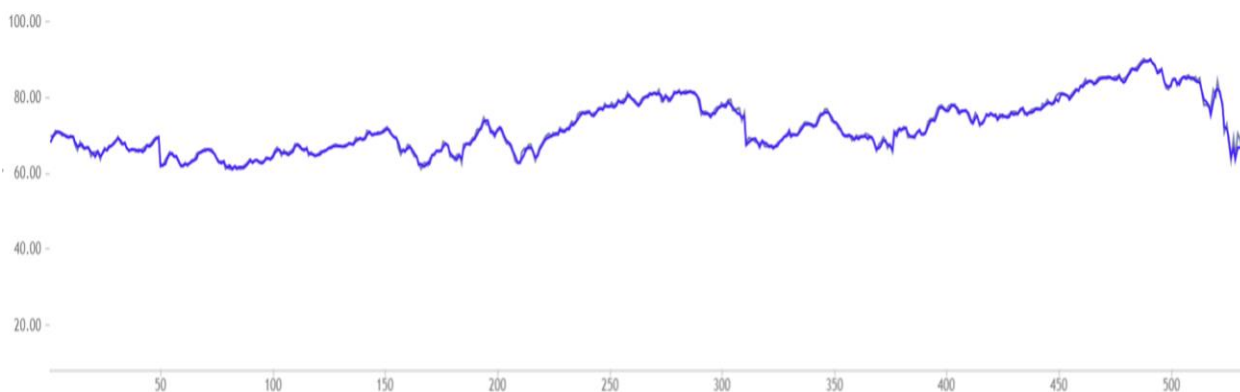


Рисунок 3.14 — Графік реальних і прогнозованих цін «А» (Agilent
Technologies)

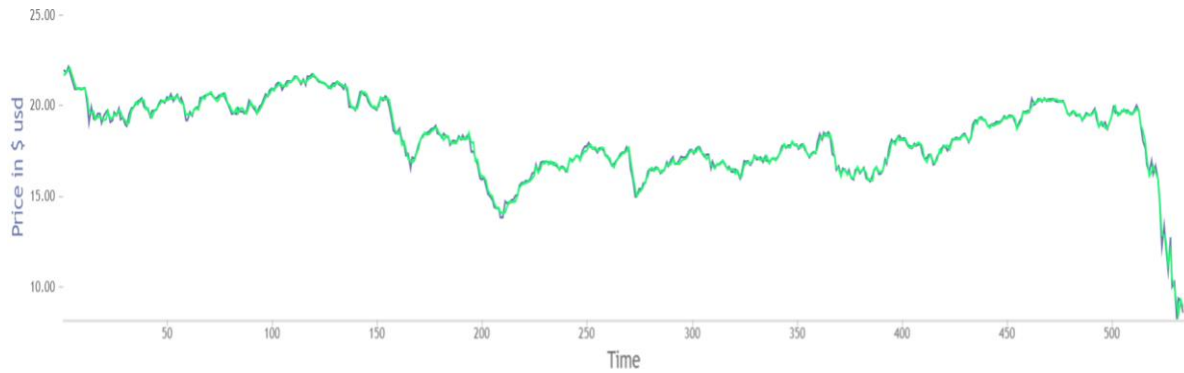


Рисунок 3.15 — Графік реальних і прогнозованих цін «KEY» (KeyCorp)

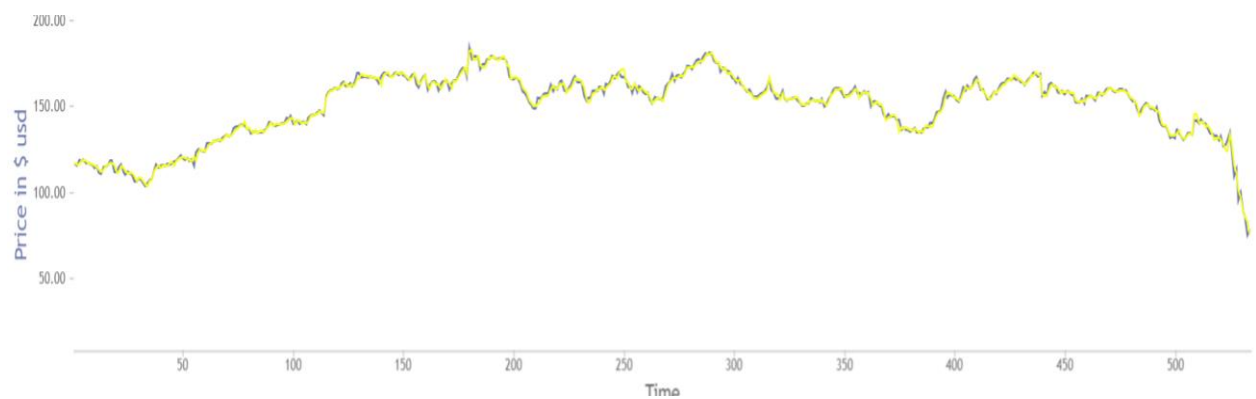


Рисунок 3.16 — Графік реальних і прогнозованих цін «AAP»
(Advance Auto Parts)

Для математичного порівняння результатів моделі з відхиленням з реальними цінами було використано дані показники:

- кореляційний коефіцієнт - це статистичний показник, що наглядно показує, наскільки пов'язані між собою певні коливання значень двох інших показників;
- середня абсолютна помилка – помилка, яка оцінює величину помилки в порівнянні зі значенням певного ряду та з помилками у сусідніх рядах векторів [30];
- середньоквадратичне відхилення - показник розсіювання значень випадкової величини щодо її математичного очікування;

– середня помилка апроксимації - середнє відхилення фактичних значень від прогнозованих, якщо дана величина дорівнює до 15%, то підібрана модель ефективна [28];

– точність моделі - це процент кількості прогнозованих даних з тестової вибірки, які були спрогнозовано вірно, тобто компанія змінила ціну протягом горизонту у 5 днів на 5 відсотків.

Таблиця 3.2 — Результати виконання моделей

Компанія	IBM	A	AAP	KEY	UPS
Точність моделі	82.35%	84.28%	81.92%	85.03%	84.21%
Кореляційний коефіцієнт	0.8576	0.8457	0.8357	0.8456	0.8324
Середня абсолютна помилка у відсотках	19.25%	16.39%	18.73%	14.32%	17.37%
Середня абсолютна помилка	23.62	32.98	46.04	3.52	36.27
Середня помилка апроксимації	15.67%	13.27%	14.29%	12.93%	13.18%
Кількість вхідних даних (торгових днів)	2532	2641	2512	2782	2398
Відсоток прибутковості при торгуванні компанією	5.89%	1.43%	2.80%	3.56%	2.81%

З отриманих результатів видно, що прогноз моделі близький до значень реальних даних. Якщо користувач користуватиметься даним автоматизованим трейдовим помічником, то він зможе отримати прибутковий інвестиційний довгостроковий портфель, яким може торгувати з мінімальними ризиками протягом довготривалого періоду.

3.7 Оцінка ефективності запропонованого рішення

Для порівняння отриманих результатів візьмемо минулорічну роботу студентки КПІ Бородай Тетяни [30], яка використовувала генеративно-змагальну нейронну мережу для прогнозування фондового ринку. Після навчання моделей їй вдалося отримати досить точний прогноз цін акцій компанії, який наближений до реальності.

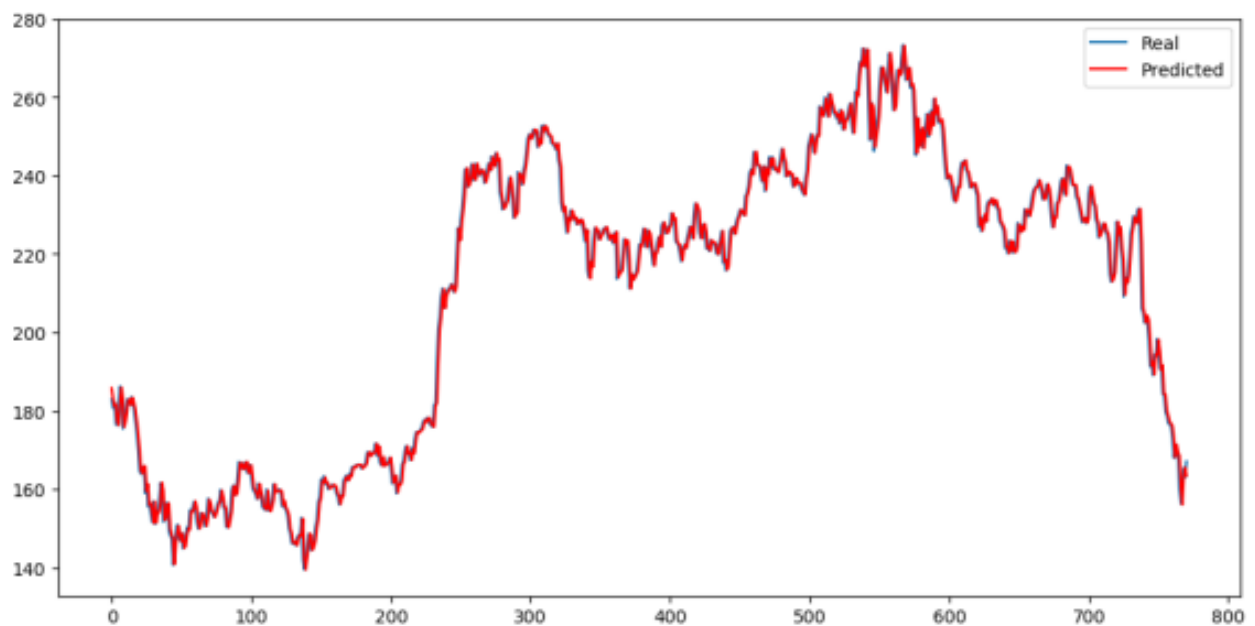


Рисунок 3.17 — Графік результатів моделі за червень 2020 рік [30]

Проте використовуючи дану модель для попередніх та сьогоднішніх випробувань, було отримано гірші результати точності. Це обумовлено тим, що модель не пристосовано до мінливості ринку. На рисунку зображено графік прогнозування цін акцій компанії за червень 2020 рік, де ефективність моделі дорівнює 81,14%. Модель близько наблизилась до реальних даних навідміну від експерименту, де вхідними параметрами були ціни за листопад 2021 рік.

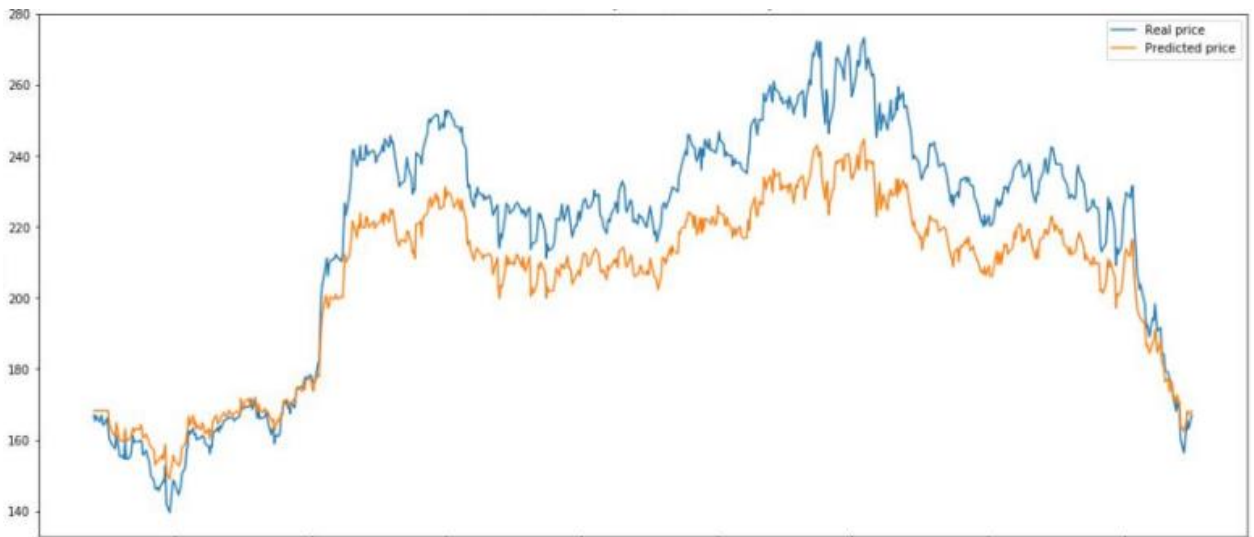


Рисунок 3.18 — Графік результатів моделі для майбутніх значень [30]

Для оцінки ефективності рішення порівняння моделей було взято середнє значення точності усіх прогонів моделі для конкретних періодів (дані періоди представлені у таблиці 3.3, у якій детально порівнюються результати виконання описаної моделі) з горизонтом у 5 днів. Подібний експеримент Для прогону генеративно-змагальна нейронної мережі було взято такі фінансові дані, як:

- волатильність;
- ціна відкриття (open price);
- ціна закриття (close price);
- максимальне значення ціни акції протягом дня (high price);
- мінімальне значення ціни акції протягом дня (low price).

Для описаної розробленої моделі на відміну від генеративно-змагальної нейронної мережі окрім вищеперелічених характеристик вище використовуються:

- новини для кожної компанії, які котируються на американських біржах (новини включають короткий опис, зображення, текст, посилання на джерело новин, тощо);

– інформація про фінансові звіти, як форма 10-K/Q – річні та поквартальні звіти (включаючи основну частину числових даних зі звіту, як-от різні типи активів та зобов'язань);

– основна інформація про кожну компанію (назва компанії, номер, адреса, місцевість, галузь, номер телефону тощо);

– детальна інформація про виконавчий персонал NYSE та NASDAQ компаній, які використовуються у прогнозуванні (посада, ім'я, вік, стать, рік затвердження тощо).

Таблиця 3.3 — Порівняння ефективності моделей

Період прогнозування	Точність розробленої моделі, %	Точність генеративно-змагальної нейронної мережі, %	Час виконання мікросервісних моделей	Час виконання монолітної паралельної моделі
01.04.2020-31.05.2020	81,73	80,21	4 хв 32 с	5 хв 56 с
01.06.2020-31.07.2020	82,36	81,14	3 хв 16 с	4 хв 12 с
01.08.2020-30.09.2020	84,55	74,32	5 хв 21 с	6 хв 34 с
01.10.2020-30.11.2020	81,51	78,19	5 хв 19 с	6 хв 2 с
01.12.2020-31.01.2021	87,25	77,88	3 хв 57 с	4 хв 29 с
01.02.2021-31.03.2021	76,24	74,55	3 хв 42 с	4 хв 53 с
01.04.2021-31.05.2021	72,59	63,78	4 хв 39 с	5 хв 42 с

Продовження таблиці 3.3

Період прогнозування	Точність розробленої моделі, %	Точність генеративно-змагальної нейронної мережі, %	Час виконання мікросервісних моделей	Час виконання монолітної паралельної моделі
01.06.2021-31.07.2021	90,45	71,25	5 хв 2 с	5 хв 32 с
01.08.2021-30.09.2021	80,12	68,71.	5 хв 21 с	6 хв 28 с
01.10.2021-30.11.2021	74,51	69,67	4 хв 58 с	6 хв 8 с
Середнє арифметичне	81,13	73,97	4 хв 37 с	5 хв 36 с

Порівнюючи результати моделі, яка реалізована у програмному забезпеченні у рамках даної магістерської дисертації, та моделі минулорічної роботи студентки КПІ Бородай було виявлено, що прогноз розробленої моделі близький до значень реальних даних протягом довгого періоду (у даному дослідженні – 1,5 роки) у порівнянні з попередніми результатами. Час виконання мікросервісних моделей на 17,55% швидше за монолітну паралельну модель.

Розроблений алгоритм на основі аналізу новин показує хорошу точність прогнозування без залежності від мінливості ринку.

3.8 Висновки до розділу

У третьому розділі були описано контролери, архітектура програмного забезпечення та кінцеві точки додатку. Було обрано засоби для реалізації математичного забезпечення та збереження, оброблення, виведення результатів виконання моделей і генерацію датасетів. Проаналізовано безпеку

даних користувача та продемонстровано головну функціональність додатку. Було проведено експериментальні дослідження для того щоб зробити оцінку ефективності запропонованого рішення.

Розроблений алгоритм на основі аналізу новин та інформації про виконавчий персонал компанії має стабільні результати точності прогнозування цін протягом тривалого періоду. Аналоги демонструють результати, у яких зменшується точність в арифметичній прогресії до реальних значень на 15-17% на терміні 1,5 роки. Описана модель має стабільну точність в середньому у 81,13% протягом усього експериментального періоду в незалежності від мінливості ринку. Час виконання мікросервісних моделей на 17,55% швидше за монолітну паралельну модель.

4 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

4.1 Опис ідеї проєкту

Таблиця 4.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмне та математичне забезпечення автоматизованого трейдингового помічника	Комплексне математичне та програмне забезпечення автоматизованого торгового помічника для електронних торгів, яке на відміну від аналогів не втрачає власну актуальність та буде прибутковою для користувача протягом довгого терміну.	Користувачі матимуть доступ до інтуїтивно-зрозумілого інтерфейсу з прогнозованими довготерміновими біржовими цінами акцій. Програмний інтерфейс для отримання та використання результатів моделі сторонніми додатками.

4.2 Технологічний аудит ідеї проєкту

Таблиця 4.2 — Опис ідеї стартап-проєкту

№	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
Мій проєкт	Немає індикаторів профіту. Немає повної торгової функціональності	Створення графічного інтерфейсу для отримання результатів. Розробка програмного інтерфейсу для отримання та використання результатів моделі сторонніми додатками. -Автоматизація користувача.	Зменшення ризиків користувачів. Відсутність залежності від ринку. Покращення ефективності прогнозування цін акцій. Оптимізація торгового портфелю.
Strategy Quant	Базова ліцензія коштує 1290 доларів США.	Автоматизоване тестування з використанням	Розробка стратегії за допомогою дуже зручного інтерфейсу і сотень

Продовження таблиці 4.2

	Стратегія залежить лише від технічного аналізу. Залежить від мінливості ринку. Немає індикаторів профіту.	випадковості і на основі широкого спектру параметрів. Вбудований оптимізатор руху вперед плюс безліч інших інструментів оптимізації (Монте-Карло і т. д.).	вбудованих стратегій. Чотири режими (побудова, повторне тестування, поліпшення, оптимізація). Включає в себе більше 40 індикаторів, патернів, свічників і т. д.
EA Builder	Ліцензія з повним функціоналом коштує 564 доларів США. Стратегія залежить лише від технічного аналізу. Залежить від мінливості ринку. Немає індикаторів профіту. Жорсткий контроль з боку керівництва (розмір, ризик, поширення і час).	Повна торгова функціональність; Створення індикаторів для безкоштовних і призначених для користувача радників; Стрілки за допомогою графічного інтерфейсу; Необмежені змінні для створення автоматичних стратегій; Повні функції управління капіталом; Кольоровий індикатор CCI і попередження про розрив тренда.	Створення індикаторів безкоштовно і без будь-яких навичок програмування. EA Builder включає в себе десятки функцій, які можуть інтегрувати ваші ідеї і перетворити їх в повністю працюючу автоматизовану систему Форекс. Працює на будь-якому комп'ютері або смартфоні. Згенеровані радники можуть працювати на різних платформах.

Продовження таблиці 4.2

Forex Trendy Scanner	Ліцензія з повним функціоналом коштує 564 доларів США. Стратегія залежить лише від технічного аналізу. Залежить від мінливості ринку. Немає індикаторів профіту.	Графіки популярних валютних пар в декількох часових періодах. Автоматичний аналіз графіків з визначенням прапорів, трикутників, ліній тренда і клинів по 34 валютним парам. Оповіщення (звукові і по електронній пошті); Короткий огляд всіх тенденцій і всіх валютних пар.	Щоденний аналіз 34 валютних пар. Живі сигнали по електронній пошті плюс звукові оповіщення. Просте налаштування валютної пари Візуалізований торговий інтерфейс. Автоматичне виявлення прапорів, ліній тренду, трикутників і клинів.
----------------------	--	---	--

Таблиця 4.3 — Технологічна здійсненність ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Збереження, зміна та видалення даних у організованих системах	PostgreSQL	Наявна	Доступна
2	Математичне забезпечення та бізнес логіка проєкту	Spring	Наявна	Доступна
3	Класифікаційна та регресійна модель для прогнозування акцій	Weka	Наявна	Доступна
4	Інтуїтивно-зрозумілий інтерфейс та API для сторонніх джерел	React	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: мова програмування Java (Spring Framework), база даних – PostgreSQL, фреймворки алгоритмів машинного навчання для класифікації та регресії – Weka, фреймворк візуального інтерфейсу – React				

4.3 Аналіз ринкових можливостей запуску стартап-проєкту

Таблиця 4.4 — Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн./ум.од	500000 доларів (з розрахунку по 500 проданих річних підписок)
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі або по ринку, %	Невідомо

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим для реалізації та просування даного програмного забезпечення.

Таблиця 4.5 — Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Програмне забезпечення автоматизованого трейдингового помічника, що не залежить від мінливості ринку	Досвідчені трейдери, трейдери початківці, крупні торгові організації, аналітики	Крупні компанії, що торгують грошима клієнтів більш вимогливі до точності та до довговічності роботи алгоритмів	Інтуїтивна зрозумілість інтерфейсу, мінімальна необхідна кількість дій для отримання результату Програмний інтерфейс для отримання та використання результатів моделі.

Таблиця 4.6 — Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість продуктом (відсутність попиту)	Вузькоспрямованість продукту, відсутність розуміння проблеми у потенційних клієнтів	Проведення більш спеціалізованої рекламної кампанії, донесення до клієнтів необхідності використання забезпечення, проведення онлайн-занять з демонстрацією використання розробленого забезпечення
2	Відсутність коштів на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту для покрокового виведення продукту на ринок та отримання відповіді користувачів

Таблиця 4.7 — Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Розвиток сфери ринку іноземних валют	Збільшення конкуренції, підвищення волатильності основних інструментів	Більш активна пропаганда (реклама) свого математичного та програмного забезпечення
2	Зростання кількості трейдерів початківців	Ринок переповнений новачками, які програють гроші з великою ймовірністю	Надання послуг цій цільовій аудиторії з метою досягнення ними позитивного досвіду при торгівлі і тим самим збереження своїх клієнтів
3	Поява нових алгоритмів і технологій	Сфера машинного навчання і аналізу даних розвивається, тому з'являються нові дослідження та алгоритми	Додавання до стеку математичного і програмного забезпечення нових алгоритмів

Продовження таблиці 4.7

№	Фактор	Зміст можливості	Можлива реакція компанії
4	Велика кількість відкритих даних	Зростання кількості сервісів, що надають можливість безкоштовно (або дуже дешево) отримати дані для дослідження і роботи програмного забезпечення	Розробка нових алгоритмів і покращення вже готових з новими даними, роль посередника (постачання даних клієнтам, яким це потрібно)
5	Відсутність повноцінного продуктуаль – тернативи	Наявні продукти не дають можливості отримати такий же результат настільки швидко і за допомогою одного й того ж програмного продукту	Забезпечення стабільності роботи існуючого функціоналу, додавання нового функціоналу

Таблиця 4.8 — Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: олігополія	На ринку доступні додатки та сервіси зі схожим функціоналом, проте повноцінного конкурента немає	Перш за все забезпечити стабільність і якість роботи базових алгоритмів і запуснути додаток для перших клієнтів, розвивати існуючий функціонал
2	Рівень конкурентної боротьби: світовий	Світовий ринок немає повноцінного конкурента	Пропозиція розробленого програмного забезпечення вже існуючій аудиторії
3	Галузева ознака: внутрішньогалузева	Розробка призначена лише для використання на ринку форекс	Підтримка і покращення існуючих функцій
4	Конкуренція за видами товарів: товарно-видова	Призначення розробки – прогнозування біржових цін акцій, які не залежать від мінливості ринку.	Якість та маркетинг програмного забезпечення

Продовження таблиці 4.8

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
5	Характер конкурентних переваг: не цінова	Для користувачів важливо отримувати якісні послуги швидко та з мінімальною затратою зусиль Відмінність – наявність всього необхідного функціоналу в одному місці	Супроводження існуючого функціоналу і взаємодія з користувачами (надання зворотного зв'язку)
6	За інтенсивністю: марочна	Для того, щоб продукт став популярним необхідна хороша репутація	Надання якісного функціоналу, який не можуть надати конкуренти, спілкування і взаємодія з користувачами

Таблиця 4.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Forex Trendy Scanner	EA Builder, StrategyQuant	Відсутні	Кількість споживачів, доступність інформації	Більша популярність через захвачені сегменти ринку
Висновки	В прямого конкурента є своя аудиторія і попит	Потенційний конкурент надає якісні послуги, але не є повним аналогом	-	Якість програмного забезпечення і зацікавленість аудиторії в ньому	Обмеження роботи через товари-замінники

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею [23].

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10 — Порівняльний аналіз сильних та слабких сторін

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Швидкість роботи алгоритмів	13		+					
2	Висока якість програмного забезпечення	12					+		
3	Відповідність потребам цільової аудиторії	18						+	
4	Забезпечення відсутності залежності від впливу ринку	20	+						

Таблиця 4.11 — Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Швидкість роботи алгоритмів	Клієнти не змушені довго чекати на результат роботи алгоритму (робота займає 1-3 секунди)
2	Висока якість програмного забезпечення	ПЗ має зручний та інтуїтивно зрозумілий інтерфейс, працює безперебійно
3	Відповідність потребам цільової аудиторії	Програмне забезпечення швидко виконує поставлену перед ним задачу і отриманий результат повністю покриває потреби користувача
4	Забезпечення відсутності залежності від впливу ринку	Алгоритми побудовані на класифікації новин провідних джерел

Таблиця 4.12 — Альтернативи ринкового впровадження стартапу

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка веб-інтерфейсу застосунку	Головний ресурс – люди та час, даний ресурс - наявний	2 місяці
2	Створення API-інтерфейсу для передачі результатів роботи алгоритмів стороннім компаніям	Головний ресурс – люди та час, даний ресурс - наявний	1 місяць
3	Реклама	Залучення власних коштів для реклами товару або пошук спонсорів	1-2 місяці
4	Презентація товару	Залучення власних коштів для реклами товару або пошук спонсорів	1-2 місяці

Таблиця 4.13 — SWOT аналіз стартап-проєкту

Сильні сторони (S): – забезпечення відсутності залежності від впливу ринку; – швидкість роботи алгоритмів; – відповідність потребам цільової аудиторії; – висока якість програмного забезпечення.	Слабкі сторони (W): – немає повної торгової функціональності; – немає індикаторів профіту.
Можливості (O): – зростання кількості трейдерів початківців; – відсутність повноцінного продукту альтернативи; – велика кількість відкритих даних.	Загрози (T): – незацікавленість продуктом (відсутність попиту); – великі кошти на розробку та підтримку продукту.

4.4 Розроблення ринкової стратегії проєкту

Таблиця 4.14 — Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Трейдери, що торгують на акційних біржах	Готові	Високий	Висока	Просто
2	Великі та середні торгово-аналітично спрямовані трейдові компанії	Потребують переговорів	Середній	Висока	Складно
3	Аналітики, що працюють самі на себе та аналітики фрілансу	Готові	Середній	Середня	Просто
Які цільові групи обрано: трейдери, аналітики					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є працівники ІТ сфери, ІТ компанії в цілому та будь-які підприємства котрі використовують програмні продукти побудовані на мові програмування Java, та використовують реляційні бази даних. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств, ІТ працівників та ІТ компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 4.15 — Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Орієнтація поточної моделі на трейдерів (досвідчених і початківців)	Стратегія концентрованого маркетингу	Трейдери, яким потрібно формування торгового портфелю	Стратегія спеціалізації
Орієнтація поточної моделі на аналітиків	Поступове захоплення ринку, акцент на якість, а не на швидкість	Обрана цільова група, що принесе максимум прибутку (ті аналітики, яким потрібні додаткові інструменти)	Стратегія спеціалізації

Таблиця 4.16 — Визначення базової стратегії конкурентної поведінки

Чи є проєкт «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари-замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх за рахунок реклами переваг даного програмного забезпечення	Ні. Основний акцент під час просування даного забезпечення – простота у використанні. Зайві ускладнення будуть недоречні	Стратегія заняття конкурентної ніші

Таблиця 4.17 — Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проєкту
Швидкість роботи, мінімальна кількість витрачених сил і часу для отримання результату	Стратегія спеціалізації	Трейдерам необхідний засіб для швидкого формування торгового портфелю, аналітикам необхідний додатковий засіб в арсенал вже наявних для надання більш якісних послуг	Доступність, швидкість, унікальність, простота

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18 — Визначення ключових переваг концепції товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Отримання списку біржових компаній, які за прогнозами змінять ціну на певний відсоток протягом певного періоду	Користувачі матимуть доступ до інтуїтивно-зрозумілого інтерфейсу з прогнозованими довготерміновими цінами акцій	Програмний інтерфейс для отримання та використання результатів моделі сторонніми додатками.
2	Забезпечення відсутності залежності від впливу ринку	Довгостроковий алгоритм, який базується на новинах	Відсутність даної функціональності у конкурентів

Таблиця 4.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Програмне та математичне забезпечення автоматизованого трейдингового помічника		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Швидкість роботи алгоритмів	1. М	1. Тх
	2. Забезпечення відсутності залежності від впливу ринку	2. М	2. Тх
	3. Оптимізація	3. М	3. Тх
	4. Дешевизна	4. М	4. Вр
	Якість: математичне та програмне забезпечення		
	Марка: НТУУ КПІ ім. Ігоря Сікорського		
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника, розробка додаткового функціоналу		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент			

Таблиця 4.20 — Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
10000 грн	15000 грн	Фізичні особи – 100000 грн	10000-20000 грн

Таблиця 4.21 — Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Надання безкоштовної (частково обмеженої) та повної версій продукту	Простота в отриманні доступу до програмного забезпечення, оплаті послуг	Середня	Безпосередня

Таблиця 4.22 — Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Орієнтація на якісний і простий у використанні продукт	Спеціалізовані блоги, сайти компаній	Простота у використанні, швидкість отримання результату	Показати, що створення портфелю займає лічені хвилини навіть для недосвідченого трейдера	«Почни торгувати вже зараз», «Розумний торговий портфель – твій портфель»

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

4.6 Висновки по розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проєкту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проєкту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проєкту було встановлено що проєкт є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проєкту та доведено доцільність подальшої імплементації проєкту.

ВИСНОВКИ

У ході виконання магістерської дисертації було детально розглянуто питання пов'язані з розробки програмного та математичного забезпечення для торгування активів на фондовому ринку протягом тривалого періоду.

На основі даних, які були отримані у процесі аналізу, сформульовано задачу розробити довготерміновий алгоритм прогнозування цін, який не залежить від мінливості ринку.

Розроблено модель з використанням набору програм машинного навчання Weka, написане на Java. Все перераховане є відкритими для загального користування та безкоштовними інструментам, що дозволить продемонструвати результати математичного забезпечення.

Було реалізовано мікросервісне програмне забезпечення автоматизованого трейдового помічника. Розроблений алгоритм на основі аналізу новин та інформації про виконавчий персонал компанії має більший відсоток точності прогнозування цін протягом тривалого періоду. Аналоги демонструють результати, у яких зменшується точність в арифметичній прогресії до реальних значень на 16,23% на терміні 1,5 роки. Описана модель має стабільну точність в середньому у 81,13% протягом усього експериментального періоду в незалежності від мінливості ринку. Час виконання мікросервісних моделей на 17,55% швидше за монолітну паралельну модель.

Наукова новизна отриманих результатів даної магістерської дисертації: удосконалено:

- архітектурне рішення програмного забезпечення автоматизованого торгового помічника на основі мікросервісної архітектури;
- модель прогнозу біржових цін на основі галузі та отриманих від достовірних джерел новин, інформації про виконавчий персонал та фінансові звіти;

– зменшення часу прогнозування акцій, використовуючи мікросервісну архітектуру.

здобуло подальший розвиток:

– підтримка декількох варіантів отримання даних – за допомоги користувацького веб- та програмного (API) інтерфейсів.

За результатами роботи над магістерською дисертацією було опубліковано статтю.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1) Automatic Strategy Generation With StrategyQuant // Trading tact – посібник для автоматизованої торгової стратегії URL:
<https://tradingtact.com/strategyquant-generation/> (дата звернення: 05.10.2021).
- 2) Building Microservices: Inter-Process Communication in a Microservices Architecture // Nginx – практичний посібник із інструментів керування Kubernetes URL: <https://www.nginx.com/blog/building-microservices-inter-process-communication/> (дата звернення: 06.09.2021).
- 3) Choosing a Microservices Deployment Strategy // Nginx – практичний посібник із інструментів керування Kubernetes URL:
<https://www.nginx.com/blog/deploying-microservices/> (дата звернення: 08.11.2021).
- 4) Cross decomposition // Scikit-learn документація по аналізу даних URL:
https://scikit-learn.org/stable/modules/cross_decomposition.html (дата звернення: 21.10.2021).
- 5) Deploying microservices // Microservices.io – документація про мікросервіси, яка підтримується автором шаблонів мікросервісів Крісом Річардсоном URL: <https://microservices.io/articles/deployment.html> (дата звернення: 13.10.2021).
- 6) Does Financial News Predict Stock Returns? // ScienceDirect - джерело рецензованих журнальних статей та розділів книг URL:
<https://www.sciencedirect.com/science/article/pii/S0927538X15300317> (дата звернення: 11.11.2021).
- 7) EA Builder Club lifetime // IExpertadvisor конструктор експертів та порадників MetaTrader URL: <https://www.iexpertadvisor.com/vts-products/> (дата звернення: 02.10.2021).
- 8) Ensemble methods // Scikit-learn документація по аналізу даних URL:
<https://scikit-learn.org/stable/modules/ensemble.html> (дата звернення: 22.10.2021).

- 9) Event-Driven Data Management for Microservices // Nginx – практичний посібник із інструментів керування Kubernetes URL: <https://www.nginx.com/blog/event-driven-data-management-microservices/> (дата звернення: 17.09.2021).
- 10) Feature selection // Scikit-learn документація по аналізу даних URL: https://scikit-learn.org/stable/modules/feature_selection.html (дата звернення: 21.10.2021).
- 11) Forex Trendy Review // Trading review – сервіс фінансових запитів URL: <https://tradingreview.net/forex-trendy-review/> (дата звернення: 11.11.2021).
- 12) Gaussian Process Regression // Scikit-learn документація по аналізу даних URL: https://scikit-learn.org/stable/modules/gaussian_process.html (дата звернення: 21.10.2021).
- 13) Impact of News on the Trend of Stock Price Change: an Analysis based on the Deep Bidirectional LSTM Model // ScienceDirect - джерело рецензованих журнальних статей та розділів книг URL: <https://www.sciencedirect.com/science/article/pii/S1877050920315829> (дата звернення: 19.10.2021).
- 14) Introduction to Microservices // Nginx – практичний посібник із інструментів керування Kubernetes URL: <https://www.nginx.com/blog/introduction-to-microservices/> (дата звернення: 22.09.2021).
- 15) Kernel ridge regression // Scikit-learn документація по аналізу даних URL: <https://scikit-learn.org/stable/modules/multiclass.html> (дата звернення: 09.11.2021).
- 16) Kernels for Gaussian Processes // Scikit-learn документація по аналізу даних URL: https://scikit-learn.org/stable/modules/gaussian_process.html#kernels-for-gaussian-processes (дата звернення: 21.10.2021).
- 17) Linear and Quadratic Discriminant Analysis // Scikit-learn документація по аналізу даних URL: https://scikit-learn.org/stable/modules/lda_qda.html (дата звернення: 08.11.2021).
- 18) Linear Models // Scikit-learn документація по аналізу даних URL:

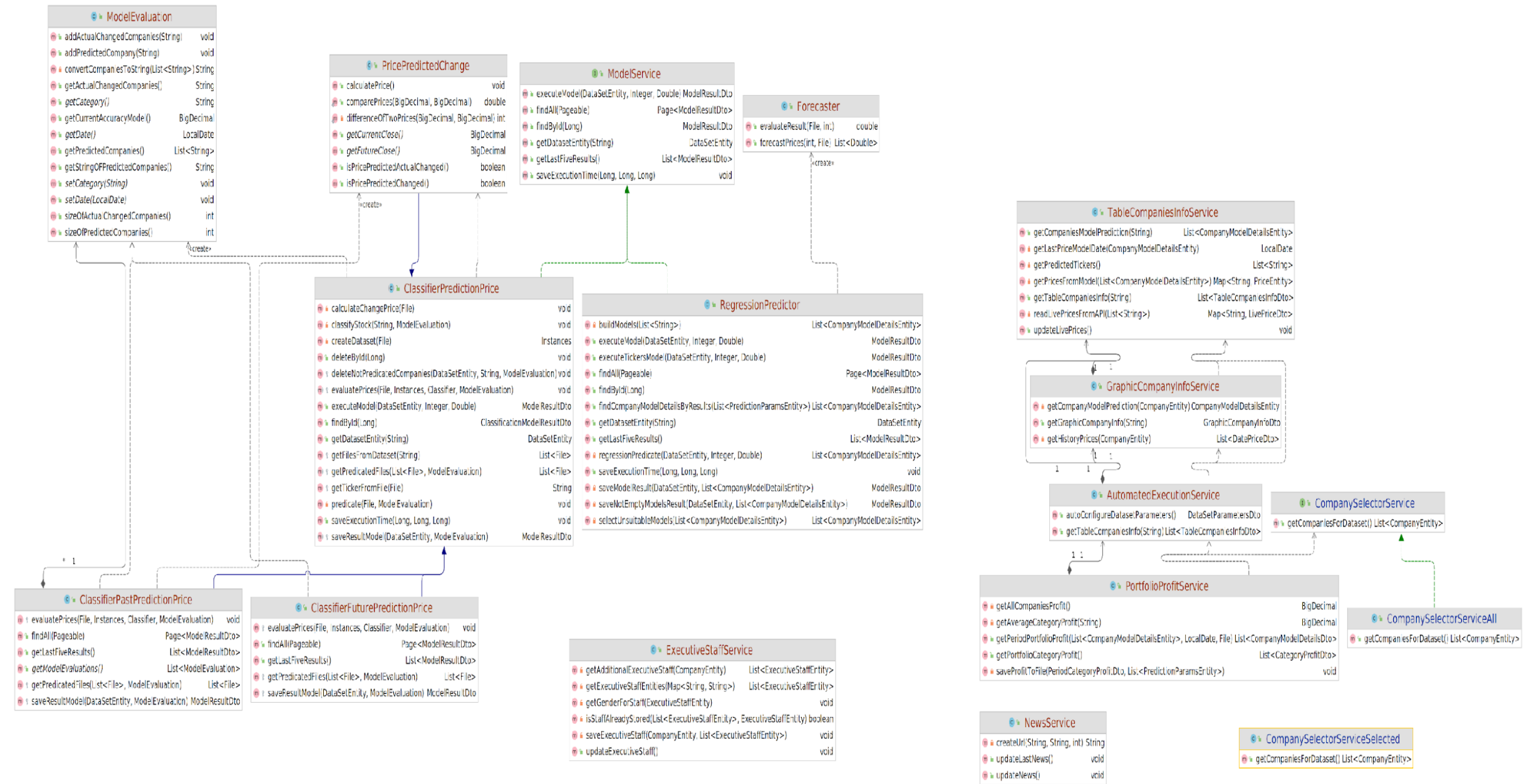
- https://scikit-learn.org/stable/modules/linear_model.html (дата звернення: 18.09.2021).
- 19) Multiclass and multioutput algorithms // Scikit-learn документація по аналізу даних URL: <https://www.iexpertadvisor.com/vts-products/> (дата звернення: 21.10.2021).
- 20) Multinomial Naive Bayes // Scikit-learn документація по аналізу даних URL: https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes (дата звернення: 07.11.2021).
- 21) New and revised microservices classes // Microservices.io – документація про мікросервіси, яка підтримується автором шаблонів мікросервісів Крісом Річардсоном URL: <https://microservices.io/microservices/general/2018/10/18/new-and-revised-classes.html> (дата звернення: 02.10.2021).
- 22) Pattern: Microservice Architecture // Microservices.io – документація про мікросервіси, яка підтримується автором шаблонів мікросервісів Крісом Річардсоном URL: <https://microservices.io/patterns/microservices.html> (дата звернення: 14.10.2021).
- 23) Predicting stock returns in the presence of COVID-19 pandemic: The role of health news // ScienceDirect - джерело рецензованих журнальних статей та розділів книг URL: <https://www.sciencedirect.com/science/article/pii/S1057521920301903> (дата звернення: 22.09.2021).
- 24) Research on stock price forecast based on gray relational analysis and ARMAX model // IEEE Xplore - технічна література найвищої якості з техніки та технологій URL: <https://ieeexplore.ieee.org/document/8077689> (дата звернення: 12.11.2021).
- 25) Service Discovery in a Microservices Architecture// Nginx – практичний посібник із інструментів керування Kubernetes URL: <https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/> (дата звернення: 09.11.2021).

- 26) Stock Price Forecasting by Combining News Mining and Time Series Analysis
// IEEE Xplore - технічна література найвищої якості з техніки та технологій
URL: <https://ieeexplore.ieee.org/document/5286063> (дата звернення: 17.10.2021).
- 27) Stock Price Forecasting Using Data from Yahoo Finance and Analysing Seasonal and Nonseasonal Trend // IEEE Xplore - технічна література найвищої якості з техніки та технологій URL:
<https://ieeexplore.ieee.org/document/8663035> (дата звернення: 01.09.2021).
- 28) The What, Why, and How of a Microservices Architecture // Medium - платформа для обміну професійним досвідом URL:
<https://medium.com/hashmapinc/the-what-why-and-how-of-a-microservices-architecture-4179579423a9> (дата звернення: 05.11.2021).
- 29) Twitter as a tool for forecasting stock market movements: A short-window event study // ScienceDirect - джерело рецензованих журнальних статей та розділів книг URL:
<https://www.sciencedirect.com/science/article/pii/S2405918817300247> (дата звернення: 21.10.2021).
- 30) Програмне та математичне забезпечення оптимізації портфелю активів на ринку іноземних валют // ELAKPI – Електронний архів наукових та освітніх матеріалів КПІ ім. Ігоря Сікорського URL:
https://ela.kpi.ua/bitstream/123456789/39929/1/Yasenova_magistr.pdf (дата звернення: 29.11.2021).

ДОДАТКИ

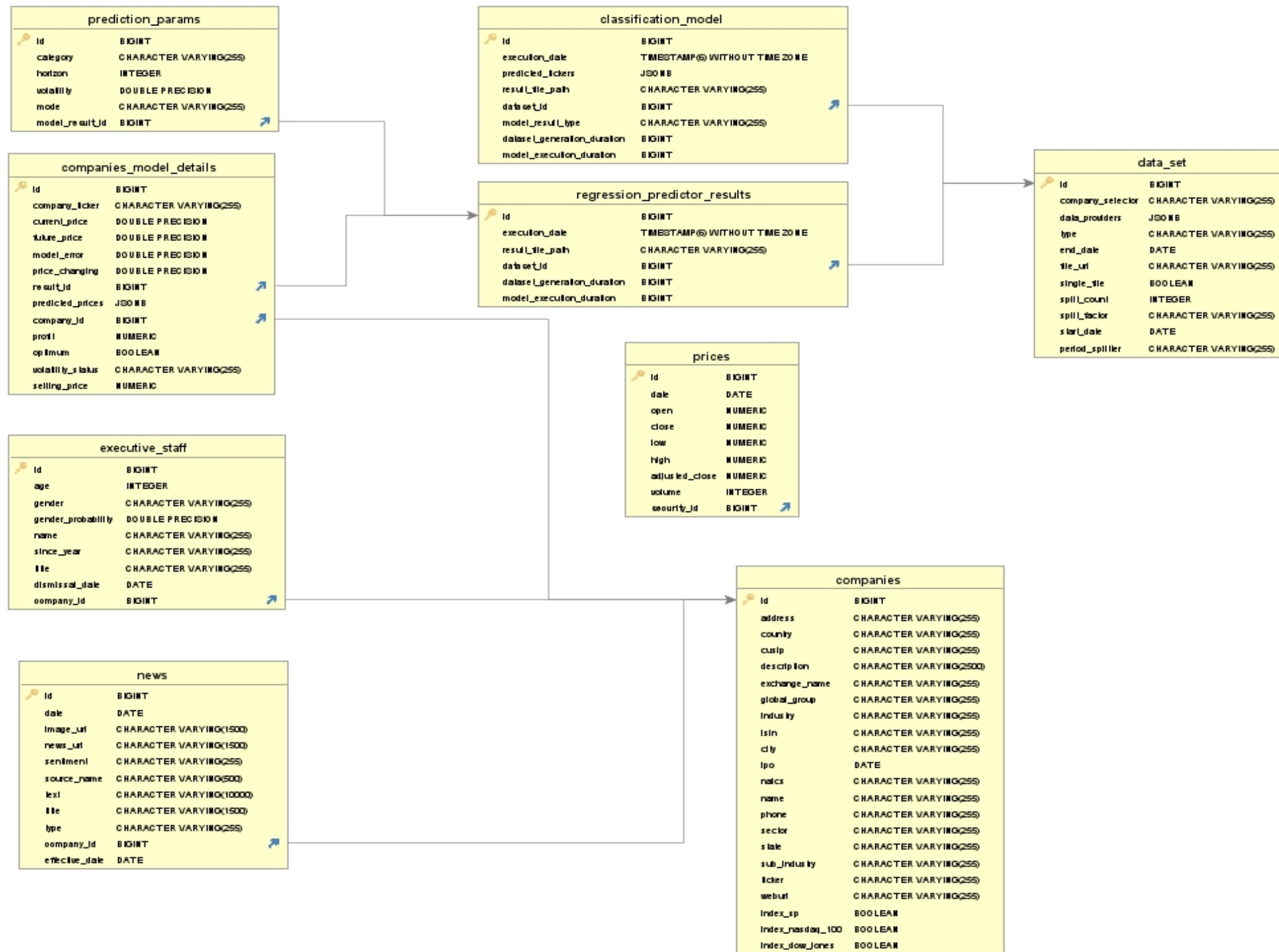
ДОДАТОК А

Діаграма класів



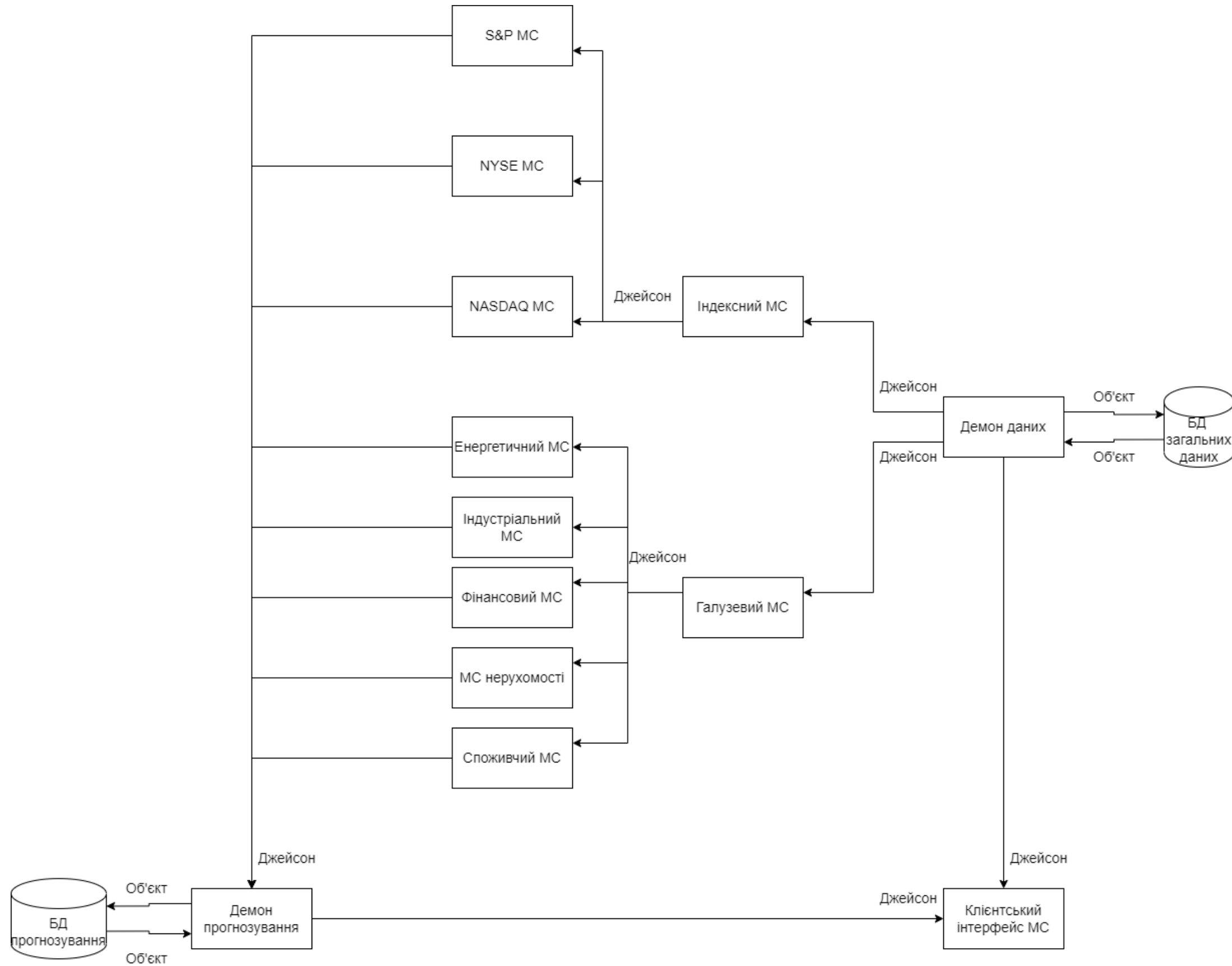
ДОДАТОК Б

Схема бази даних



ДОДАТОК В

Структурна схема архітектури програмного забезпечення



ДОДАТОК Г

Лістинг програми

PricePredictedChange.java

```
@Log4j2
@Service
@AllArgsConstructor(onConstructor = @__(@Autowired))
public class PricePredictedChange {
    private double predicted;
    private boolean actual;
    private final Instances dataset;
    private final Classifier classifier;

    public void forecastPrices() throws Exception {
        int predictedColumn = dataset.numAttributes() - 4;
        Instance daysTemp = dataset.instance(dataset.numInstances() - days);
        Instance lastTemp = dataset.instance(dataset.numInstances());
        BigDecimal currentTemp = BigDecimal.valueOf(daysTemp.value(predictedColumn));
        BigDecimal futureTemp = BigDecimal.valueOf(lastTemp.value(predictedColumn));
        actual = comparePrices(currentTemp, futureTemp);
        predicted = classifier.classifyInstance(daysTemp);
        int countShare = BigDecimal.valueOf(30000.0).divide(currentTemp, RoundingMode.UP).intValue();
        current = currentTemp.divide(BigDecimal.valueOf(countShare), RoundingMode.UP);
        future = futureTemp.divide(BigDecimal.valueOf(countShare), RoundingMode.UP);
    }

    public boolean comparePrices(BigDecimal currentClose, BigDecimal futureClose) {
        return futureClose.multiply(BigDecimal.valueOf(100))
            .divide(currentClose, RoundingMode.UP).subtract(BigDecimal.valueOf(100))
            .compareTo(percent) < 0;
    }

    public boolean isChangedPredicted() {
        return predicted == 1.0;
    }

    public boolean isActualChanged() {
        return actual;
    }
}
```

ModelComparing.java

```
@Log4j2
@Service
@AllArgsConstructor(onConstructor = @__(@Autowired))
public class ModelComparing {
    private final List<String> actualCompanies;
    private final List<String> predictedCompanies;

    public List<String> getPredictedTemp() {
        return predictedCompanies;
    }

    public void addPredicted(String predictedCompany) {
        predictedCompanies.add(predictedCompany);
    }

    public String getPredicted() {
        return convertCompaniesToString(predictedCompanies);
    }

    public String getActual() {
```

```

    return convertCompaniesToString(actualCompanies);
}

public void addActual(String company) {
    actualCompanies.add(company);
}

public BigDecimal findAccuracy() {
    return !predictedCompanies.isEmpty() ? BigDecimal.valueOf(actualCompanies.size())
        .divide(BigDecimal.valueOf(predictedCompanies.size()), RoundingMode.UP)
        .multiply(BigDecimal.valueOf(6.0))
        .setScale(2, RoundingMode.HALF_EVEN) : BigDecimal.ZERO;
}

public int predictedSize() {
    return predictedCompanies.size();
}

public int actualSize() {
    return actualCompanies.size();
}
}

```

LastPrediction.java

```

@Log4j2
@Service
@AllArgsConstructor(onConstructor = @__(@Autowired))
public class LastPrediction extends PredictionPrice {

    protected void comparingPrices(File stockFile, Instances test, Classifier classifier, ModelComparing modelComparing) throws
    Exception {
        PricePredictedChange pricePredictedChange = new PricePredictedChange(test, classifier);
        pricePredictedChange.forecastPrices();
        String ticker = parseTickers(stockFile);
        if (pricePredictedChange.isChangedPredicted()) {
            modelComparing.addPredicted(ticker);
            if (pricePredictedChange.isActualChanged()) {
                modelComparing.addActual(ticker);
            }
        }
    }

    @Override
    public List<ModelResult> getLastTenResults() {
        return new ArrayList<ModelResult>(modelRepository.findLastFiveResults(PageRequest.of(0, 10),
        ModelResultType.PREDICTED));
    }

    @Override
    public Page<ModelResult> findAll(Pageable pageable) {
        return modelRepository.findAllByModelResultType(pageable, ModelResultType.PREDICTED);
    }

    @Override
    protected ModelResult saveResultModel(Dataset dataset, ModelComparing modelComparing) {
        List<String> copyCompanies = new ArrayList<>(modelComparing.getPredicted());
        J48ModelResult j48ModelResult = J48ModelResult.builder().predictedTickers(copyCompanies)

        .resultFilePath(priceParseUtil.savePastResult(modelComparing)).dataset(dataset).modelResultType(ModelResultType.PREDI
        CTED)
        .build();
        return modelRepository.save(j48ModelResult);
    }

    @Override

```

```

protected List<File> convertPredicatedFiles(List<File> filesForDataset, ModelComparing modelComparing) {
    return filesForDataset.stream().filter(file -> modelComparing.getPredicted().contains(parseTickers(file)))
        .collect(Collectors.toList());
}
}

```

ActualForecasting.java

@Log4j2

@Service

@AllArgsConstructor(onConstructor = @__(@Autowired))

```

public class ActualForecasting extends PredictionPrice {
    private final List<String> predictedChangeCompanies;

```

```

    @Override
    public List<ModelResult> getLastTenResults() {
        return new ArrayList<ModelResult>(modelRepository.findLastFiveResults(PageRequest.of(0, 10),
            ModelResultType.ACTUAL));
    }

```

```

    @Override
    public Page<ModelResult> findAll(Pageable pageable) {
        return modelRepository.findAllByModelResultType(pageable, ModelResultType.ACTUAL);
    }

```

```

    @Override
    protected ModelResult saveResultModel(Dataset dataset, ModelComparing modelComparing) {
        List<String> copyCompanies = new ArrayList<>(predictedChangeCompanies);
        J48ModelResult j48ModelResult = J48ModelResult.builder().predictedTickers(copyCompanies)
            .resultFilePath(priceParseUtil.saveFutureResult(predictedChangeCompanies))
            .dataset(dataset).modelResultType(ModelResultType.ACTUAL).build();
        modelRepository.save(j48ModelResult);
        predictedChangeCompanies.clear();
        return modelMapper.fromEntityToDto(j48ModelResult);
    }

```

```

    protected void comparingPrices(File stockFile, Instances testDataset, Classifier model, ModelComparing modelComparing)
        throws Exception {
        Instance lastInstance = testDataset.instance(testDataset.numInstances() - 1);
        double priceChangePredicted = model.classifyInstance(lastInstance);
        String parseTickers = parseTickers(stockFile);
        if (priceChangePredicted == 1.0) {
            predictedChangeCompanies.add(parseTickers);
        }
    }
}

```

```

    @Override
    protected List<File> convertPredicatedFiles(List<File> filesForDataset, ModelComparing modelComparing) {
        return filesForDataset.stream().filter(file -> predictedChangeCompanies.contains(parseTickers(file)))
            .collect(Collectors.toList());
    }
}

```

GaussForecaster.java

@Slf4j

@Service

```

public class GaussForecaster {
    public List<Double> forecastPrices(int days, File file) throws Exception {
        CSVLoader csvLoader = new CSVLoader();
        csvLoader.setSource(file);
        Instances dataset = csvLoader.getDataSet();
        WekaForecaster prediction = new WekaForecaster();
        prediction.getTSLagMaker().setMaxLag(100);
        prediction.getTSLagMaker().setMinLag(7);
        prediction.buildForecaster(dataset);
    }
}

```

```

prediction.setBaseForecaster(new GaussianProcesses());
prediction.primeForecaster(dataset);
prediction.setFieldsToForecast("price");
return prediction.forecast(days).stream().map(p -> p.get(0).predicted()).collect(Collectors.toList());
}

public double resultEvaluating(File formattedData, int forecastForDays) throws Exception {
    FileReader reader = new FileReader(formattedData);
    CSVReader csv = new CSVReader(reader);
    List<String[]> csvRead = csv.readAll();
    List<Double> errors = new ArrayList<>();
    double realPrice = Double.parseDouble(csvRead.get(csvRead.size() - 1)[0]);
    csvRead.forEach(csvRead.remove(csvRead.size() - 1));
    List<Double> predictedPrices = forecastPrices(forecastForDays, file);
    double predictedPrice = predictedPrices.get(predictedPrices.size() - 1);
    errors.add(Math.abs(realPrice - predictedPrice) / realPrice);
    return errors.stream().mapToDouble(Double::doubleValue).average().orElse(Double.NaN);
}
}

```

PredictionPrice.java

```

@Log4j2
@NoArgsConstructor
@Service
public abstract class PredictionPrice implements PredictionModelService {
    protected ModelRepository modelRepository;
    protected ModelComparing modelComparing;
    protected PriceParseUtil priceParseUtil;
    protected FileUtil fileUtil;

    @Override
    public ModelResult run(Dataset dataset, Integer days, Double changing) throws IOException {
        Constants.setPercent(BigDecimal.valueOf(changing));
        Constants.setdays(days);
        priceParseUtil.prepareDatasetOfModel(dataset.getResultFilePath());
        modelComparing.setCategory(dataset.getCompanySelector());
        modelComparing.setDate(dataset.getEndPeriod());
        File datasetOfModel = priceParseUtil.prepareDatasetOfModel(dataset.getResultFilePath());
        List<File> predicatedFiles = convertPredicatedFiles(datasetOfModel, modelComparing);
        Stream<Path> walk = Files.walk(Paths.get(datasetOfModel));
        walk.filter(Files::isRegularFile).map(Path::toFile).collect(Collectors.toList()).forEach(fileUtil::deleteFile);
        predicatedFiles.forEach(fileUtil::deleteFile);
        return saveResultModel(dataset, modelComparing);
    }

    @Override
    public Dataset getDatasetEntity(String resultFilePath) {
        return modelRepository.findByResultFilePath(resultFilePath).get().getDataset();
    }

    protected abstract ModelResult saveResultModel(Dataset dataset, ModelComparing modelComparing);

    private void forecasting(String dataSetDirectory, ModelComparing modelComparing) throws IOException {
        Stream<Path> walk = Files.walk(Paths.get(dataSetDirectory));
        List<File> stocks = walk.filter(Files::isRegularFile).map(Path::toFile).collect(Collectors.toList());
        stocks.forEach(stock -> {
            List<Price> price = priceParseUtil.readPricesFile(stock);
            price.get(price.size() - 1);
            price.get(0);
            J48 model = new J48();
            List<Price> redPrices = priceParseUtil.readPricesFile(stock);
            for (int i = 0; i < redPrices.size() - days; i++) {
                BigDecimal current = redPrices.get(i).getClose();
            }
        });
    }
}

```

```

        BigDecimal future = redPrices.get(i + days).getClose();
        redPrices.get(i).setChange(PricePredictedChange.comparePrices(current, future));
    }
    priceParseUtil.savePrices(redPrices, price);
    CSVLoader loader = new CSVLoader();
    loader.setSource(price);
    Instances dataset = loader.getDataSet();
    dataset.setClassIndex(dataset.numAttributes() - 1);
    int splitter = (int) (dataset.numInstances() * 0.8);
    Instances test = new Instances(dataset, splitter, dataset.numInstances() - splitter);
    Instances train = new Instances(dataset, 1, splitter);
    comparingPrices(price, test, model, modelComparing);
    model.buildClassifier(train);
    dataset.clear();
});
}

@Override
public void saveTime(Long datasetGenerationDuration, Long modelExecutionDuration, Long modelResultId) {
    J48ModelResult modelResult = modelRepository.findById(modelResultId);
    modelRepository.save(modelResult.toBuilder()
        .datasetGenerationDuration(datasetGenerationDuration)
        .modelExecutionDuration(modelExecutionDuration)
        .build());
}

protected abstract void comparingPrices(File stockFile, Instances testDataset, Classifier model, ModelComparing
modelComparing)
    throws Exception;

protected abstract List<File> convertPredicatedFiles(List<File> filesForDataset, ModelComparing modelComparing);

protected String parseTickers(File stockFile) {
    return stockFile
        .getName()
        .substring(0, stockFile.getName().lastIndexOf('.'));
}
}

```

RegressionPredictor.java

```

@Slf4j
@Service
@AllArgsConstructor(onConstructor = @__(@Autowired))
public class RegressionPredictor implements PredictionModelService {
    private final RegressionRepository regressionRepository;
    private final RegressionMapper regressionMapper;
    private final PriceParseUtil priceParseUtil;
    private final CompanyDetailsRepository companyDetailsRepository;

    @Override
    public ModelResult run(Dataset dataset, Integer days, Double changing) throws IOException {
        List<CompanyDetails> models = regressionPrices(dataset, days, changing);
        return saveResultOfPrediction(dataset, models);
    }

    public ModelResult runTickersModel(Dataset dataset, Integer days, Double changing) throws IOException {
        List<CompanyDetails> models = regressionPrices(dataset, days, changing);
        return saveResultOfPrediction(dataset, models);
    }

    private ModelResult saveResultOfPrediction(Dataset dataset, List<CompanyDetails> models) {
        RegressionResult result =
            regressionRepository.save(RegressionResult.builder().executionDate(LocalDateTime.now()).dataset(dataset).build());
        return regressionMapper.fromEntityToDto(result);
    }
}

```

```

}

private List<CompanyDetails> regressionPrices(Dataset dataset, Integer days, Double changing) throws IOException {
    Constants.setdays(days);
    Constants.setPercent(BigDecimal.valueOf(changing));
    Constants.setFixes(changing);
    File zipFile = priceParseUtil.prepareDatasetOfModel(dataset.getResultFilePath());
    return buildModels(companies);
}

private ModelResult saveNotEmpty(Dataset dataset, List<CompanyDetails> models) {
    RegressionResult result =
regressionRepository.save(RegressionResult.builder().executionDate(LocalDateTime.now()).dataset(dataset).build());
    models.forEach(model -> companyDetailsRepository.save(model.toBuilder().result(result).build()));
    return regressionMapper.fromEntityToDto(result);
}

private List<CompanyDetails> buildModels(List<String> companies) {
    List<CompanyDetails> models = new ArrayList<>();
    for (String companyTicker : companies) {
        List<Double> predictedPrices = predictor.forecastPrices(days);
        double currentPrice = predictor.getCurrentPriceForCompany(companyTicker);
        double futurePrice = predictor.get(predictedPrices.size() - 1);
        double modelError = predictor.evaluateResult(days);
        double priceChanging = predictor(futurePrice - currentPrice) / currentPrice;
        models.add(companyModelDetails);
    }
    return models;
}

private List<CompanyDetails> selectModels(List<CompanyDetails> models) {
    return models.stream().filter(model -> ADMISSIBLE_MODEL_ERROR < model.getModelError() || fixes >
Math.abs(model.getPriceChanging())
        || MAX_PRICE_CHANGING < Math.abs(model.getPriceChanging()))).collect(Collectors.toList());
}

@Override
public List<ModelResult> getLastTenResults() {
    return regressionRepository.findLastFiveResults(PageRequest.of(0, 5)).stream()
        .map(regressionMapper::fromEntityToDto).collect(Collectors.toList());
}

@Override
public ModelResult findById(Long id) {
    return regressionRepository.findById(id).map(regressionMapper::fromEntityToDto);
}

@Override
public Page<ModelResult> findAll(Pageable pageable) {
    return regressionRepository.findAll(pageable).map(regressionMapper::fromEntityToDto);
}

@Override
public void saveTime(Long datasetTime, Long modelTime, Long modelResultId) {
    RegressionResult modelResult = regressionRepository.findById(modelResultId);
    regressionRepository.save(modelResult.toBuilder()
        .datasetTime(datasetTime)
        .modelTime(modelTime)
        .build());
}

public List<CompanyDetails> findCompanyModelDetailsByResults(List<PredictionParams> results) {
    return companyDetailsRepository.findAllByResultIn(results.stream()
        .map(PredictionParams::getRegressionPredictorResult)

```

```

        .collect(Collectors.toList());
    }
}

```

PriceParseUtil.java

```

@Slf4j
@AllArgsConstructor(onConstructor = @__(@Autowired))
public class PriceParseUtil {
    private final FileUtil fileUtil;

    public String savePastResult(ModelComparing modelComparing) throws IOException {
        File resultFile = new File(RESULT_DIRECTORY);
        Writer writer = Files.newBufferedWriter(resultFile.toPath());
        writePastResult(modelComparing, writer);
        fileUtil.saveFile(resultFile);
        return resultFile.getName();
    }

    private void writePastResult(ModelComparing modelComparing, Writer writer) throws IOException {
        writer.write(modelComparing.getPredicted() + "\n");
        writer.write(modelComparing.getActual() + "\n");
        writer.write(modelComparing.findAccuracy() + "%\n");
    }

    public String saveFutureResult(List<String> companies) throws IOException {
        File resultFile = new File(RESULT_DIRECTORY);
        Writer writer = Files.newBufferedWriter(resultFile.toPath());
        CSVWriter csvWriter = new CSVWriter(writer);
        companies.forEach(ticker -> csvWriter.writeNext(new String[] {ticker}));
        fileUtil.saveFile(resultFile);
        return resultFile.getPath();
    }

    public void savePrices(List<Price> predicated, File companyPrices) throws IOException {
        Writer writer = Files.newBufferedWriter(companyPrices.toPath());
        CSVWriter csvWriter = new CSVWriter(writer);

        String[] titles = {"open", "high", "low", "close"};
        csvWriter.writeNext(titles);
        predicated.forEach(predicatePrice -> csvWriter.writeNext(new String[] {String.valueOf(predicatePrice.getClose()),
            String.valueOf(predicatePrice.getOpen()), String.valueOf(predicatePrice.getHigh()),
            String.valueOf(predicatePrice.getLow())
        })));
    }

    public List<Price> readPricesFile(File companyPricesFile) throws IOException {
        CSVReader reader = new CSVReader(new FileReader(companyPricesFile));
        String[] line;
        List<Price> result = new ArrayList<>();
        while ((line = reader.readNext()) != null) {
            result.add(Price.builder().close(new BigDecimal(line[4])).open(new BigDecimal(line[1])).high(new BigDecimal(line[2]))
                .low(new BigDecimal(line[3])).build());
        }
        return result;
    }

    public File prepareDatasetOfModel(String archive) throws IOException {
        InputStream datasetStream = fileUtil.getFileFromFtpServer(archive);
        File file = new File(UUID.randomUUID().toString());
        FileUtils.copyInputStreamToFile(datasetStream, file);
        return file;
    }
}

```

```
public void convert(InputStream inputStream, File file) throws IOException {  
    FileUtils.copyInputStreamToFile(inputStream, file);  
}  
}
```

ДОДАТОК Д

Перевірка на антиплагіат



Имя пользователя:
Лісовиченко Олег Іванович

ID проверки:
1009386992

Дата проверки:
28.11.2021 15:42:57 EET

Тип проверки:
Doc vs Internet + Library

Дата отчета:
28.11.2021 15:48:54 EET

ID пользователя:
76913

Название файла: IT-04мп_Зайченко_Василь_ПЗ

Количество страниц: 47 Количество слов: 10234 Количество символов: 77181 Размер файла: 98.73 KB ID файла: 1009406805

0.66% Совпадения

Наибольшее совпадение: 0.25% с источником из Библиотеки (ID файла: 8186589)

0.26% Источники из Интернета	52	Страница 49
0.65% Источники из Библиотеки	60	Страница 49

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников