

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки

«Затверджую»  
Зав. кафедри теоретичної та  
прикладної системотехніки  
д.т.н., проф. С. І. Шматков  
«\_\_» \_\_\_\_\_ 2024 р

## Пояснювальна записка

до кваліфікаційної роботи  
бакалавра

на тему: «МОДЕЛЬ ПРОГНОЗУВАННЯ ЦІН НА ТОВАРИ НА ОСНОВІ  
МЕТРИК СОЦІАЛЬНИХ МЕРЕЖ»

Захищено на засіданні  
Атестаційної комісії № 42  
протокол № \_\_ від \_\_.06.2024 р.  
Оцінка \_\_\_\_ / \_\_\_\_  
Голова Атестаційної комісії

\_\_\_\_\_  
**СКОБ Ю. О.**

Виконав:  
студент 4 курсу, групи КІ-41  
Галузь знань: 12 – Інформаційні  
технології  
Спеціальність: 123 – Комп'ютерна  
інженерія.

**Білоус Андрій Миколайович**  
**Керівник:**

канд. екон. наук, доц. кафедри  
теоретичної та прикладної

**Чуб Ольга Ігорівна**

**Рецензент:** д.т.н., професор, професор  
кафедри  
теоретичної та прикладної інформатики  
**ФРОЛОВ В'ячеслав Вікторович**

Харків – 2024

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. Загальний обсяг роботи складає 85 сторінок, із яких 41 сторінок основної частини з 21 рисунками, 26 найменуваннями списку використаних джерел та трьома додатками.

Метою кваліфікаційної роботи є підвищення прибутковості підприємства, яке здійснює торгівлю споживчими товарами через мережу Інтернет..

**Об'єкт дослідження** – процес ціноутворення на цифровому ринку споживчих товарів.

**Предмет дослідження** – методи та інструменти прогнозування цін на споживчі товари на основі метрик соціальних мереж.

Проблема, яка вирішується в кваліфікаційній роботі, полягає в ефективному використанні даних соціальних мереж для прогнозування цін, що дозволяє підвищити точність цінових стратегій підприємств.

Область застосування – електронна комерція та маркетинг. Розроблена модель може широко використовуватися підприємствами для адаптації цінових стратегій у режимі реального часу на основі аналізу даних соціальних мереж.

**Ключові слова:** прогнозування цін, соціальні мережі, аналітика, машинне навчання, електронна комерція, дані, Python, модель, інтерфейс, аналіз даних, маркетинг, метрики.

## **ABSTRACT**

An explanatory note to the bachelor's qualification work is created in the introduction, three sections, conclusions, a list of sources used, and three appendices. The total volume of work is 85 pages, of which 41 pages are the main part with 21 figures, 26 names in the list of used sources, and three appendices.

The aim of the thesis is the development of a price prediction model for goods based on social media metrics.

The problem solved in the diploma collaboration is the effective use of social media data for price prediction, which allows improving the accuracy of pricing strategies for enterprises.

Scope – e-commerce and marketing. The developed model can be widely used by enterprises to adapt pricing strategies in real-time based on social media data analysis.

**Keywords:** price prediction, social media, analytics, machine learning, e-commerce, data, Python, model, interface, data analysis, marketing, metrics.

## ЗМІСТ

<b>ВСТУП</b> .....	6
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ, МОДЕЛІ ТА МЕТОДИ ПРОГНОЗУВАННЯ ЦІН НА ТОВАРИ З ВИКОРИСТАННЯМ МЕТРИК СОЦІАЛЬНИХ МЕРЕЖ</b> .....	8
1.1. Алгоритми збору інформації у соціальних мережах.....	8
1.2. Застосування соціальних мереж у моделях прогнозування цін.....	9
1.3. Характеристика інструментарію прогнозування цін .....	10
1.4 Аналіз методів прогнозування цін на товари за допомогою метрик соціальних мереж.....	12
Висновки за розділом 1 .....	13
<b>РОЗДІЛ 2. МОДЕЛЮВАННЯ СТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОГНОЗУВАННЯ ЦІН</b> .....	16
2.1. Засоби агрегації неструктурованих даних .....	16
2.2. Обґрунтування вибору середовища розробки програмної системи.....	17
2.3. Побудова UML-діаграм предметної області .....	20
2.4 Розробка архітектури програмної системи .....	25
Висновки за розділом 2 .....	27
<b>РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРОГНОЗУВАННЯ ЦІН</b>	29
3.1. Програмна реалізація методу прогнозування цін на товари на основі метрик соціальних мереж.....	29
3.2. Експериментальне дослідження прикладного аспекту ціноутворення за допомогою системи прогнозування .....	36

3.3. Практичні рекомендації для користувачів системи прогнозування в частині виявлення тенденцій у цінній динаміці .....	40
Висновки за розділом 3 .....	41
<b>ВИСНОВКИ</b> .....	43
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	44
<b>ДОДАТКИ</b> .....	47

## ВСТУП

У сучасному світі соціальні мережі відіграють все більшу роль, як джерела інформації, вони також виступають каналами комунікації між користувачами з усього світу. Об'єм даних, що генерується в результаті цього може бути використано для різних цілей. Однією з них може бути аналіз метрик соціальних мереж для прогнозування цін на товари.

Прогнозування цін є важливим завданням для більшості сфер діяльності. Точні та швидкі прогнози цін допомагають підприємствам краще конкурувати на ринку. Це досягається завдяки можливості оптимізації стратегії ціноутворення та плануванню запасів.

У цій роботі досліджується потенціал використання соціальних мереж для прогнозування цін на товари чи послуги. Метриками соціальних мереж можуть виступати: коментарі, лайки, репости. Інформація про ці метрики буде слугувати джерелом даних для розробки моделі прогнозування цін.

У роботі також буде проводитись аналіз моделей та методів що застосовують аналіз соціальних мереж для прогнозування. В результаті чого буде визначено, який метод є найефективнішим та найточнішим.

Ця робота має потенціал стати важливим внеском у розвиток аналітичних інструментів для прогнозування цін на товари, що може мати значний вплив на ринок, зокрема на його ціни та маркетингові стратегії.

**Метою роботи** є розробка та апробація моделі прогнозування цін на товари на основі метрик соціальних мереж.

**Предметом дослідження** є вивчення можливостей використання даних з соціальних мереж для прогнозування цін на товари.

### **Завдання дослідження:**

1. Провести аналіз прогнозування цін на основі метрик соціальних мереж.

2. Зібрати та обробити дані з різних соціальних мереж, які пов'язані з певним брендом.

3. Розробити модель прогнозування ціни на товар на основі метрик соціальних мереж.

4. Тестування розробленої моделі на точність та ефективність.

**Актуальність дослідження** полягає у тому, що воно допоможе зрозуміти вплив соціальних мереж на ціноутворення. У зв'язку із зростанням впливу соціальних мереж на споживачів, такий підхід до ціноутворення допоможе підприємствами вдосконалити свої стратегії маркетингу та ціноутворення.

**Новизна дослідження** полягає в тому, що використовується підхід, який базується на даних з соціальних мереж. Дані з соціальних мереж надають унікальну інформацію про споживачів та їх уподобання, що є корисним для прогнозування ринкового попиту та цінових коливань.

## РОЗДІЛ 1

# ТЕОРЕТИЧНІ ЗАСАДИ, МОДЕЛІ ТА МЕТОДИ ПРОГНОЗУВАННЯ ЦІН НА ТОВАРИ З ВИКОРИСТАННЯМ МЕТРИК СОЦІАЛЬНИХ МЕРЕЖ

### 1.1 Алгоритми збору інформації у соціальних мережах

Алгоритми збору інформації у соціальних мережах відіграють важливу роль у сучасному інформаційному суспільстві. Вони забезпечують ефективний доступ до великого обсягу даних, що генеруються користувачами. Першим кроком у процесі збору інформації є виявлення релевантних джерел, таких як публікації, коментарі, та профілі користувачів. Це здійснюється за допомогою різних методів, включаючи ключові слова, хештеги та геолокаційні дані.

Одним із основних підходів є використання веб-скрейпінгу, що передбачає автоматизований збір інформації з веб-сайтів. Це включає програмне забезпечення, яке здатне імітувати дії користувача, переходити по посиланнях, та витягувати необхідну інформацію з веб-сторінок. Скрейпінг вимагає врахування структурних особливостей веб-ресурсів, таких як HTML-код та API, що забезпечує доступ до певних даних.

Ще одним важливим аспектом є аналіз природної мови (NLP), який дозволяє обробляти та інтерпретувати текстову інформацію. Алгоритми NLP використовуються для визначення тематики публікацій, виявлення емоційного забарвлення коментарів, а також для класифікації та категоризації контенту. Завдяки машинному навчанню та глибокому навчанню, ці алгоритми постійно вдосконалюються, що підвищує точність і релевантність отриманих результатів.

Соціальні мережі надають доступ до різноманітних метаданих, які можуть включати інформацію про користувачів, їхні соціальні зв'язки, взаємодії та вподобання. Використання таких даних дозволяє створювати комплексні моделі

поведінки користувачів, що є корисним для цільового маркетингу, прогнозування трендів та інших комерційних і дослідницьких завдань.

Для збору інформації використовуються також алгоритми машинного навчання, які здатні автоматично класифікувати та сортувати дані. Вони можуть аналізувати великі обсяги інформації, виявляти закономірності та тренди, що дозволяє робити більш обґрунтовані висновки. Наприклад, кластери даних можуть вказувати на виникнення нових тенденцій або зміну настроїв серед певних груп користувачів.

Не менш важливим є питання етики та конфіденційності під час збору інформації. Використання алгоритмів має враховувати права користувачів на приватність, а також відповідати законодавству щодо захисту персональних даних. Компанії та дослідники зобов'язані дотримуватися етичних норм, забезпечувати прозорість своїх методів та інформувати користувачів про те, як їхні дані збираються і використовуються.

Таким чином, алгоритми збору інформації у соціальних мережах є потужним інструментом для аналізу та інтерпретації великих обсягів даних. Вони дозволяють отримувати цінну інформацію, яка може бути використана в різних галузях, від маркетингу до наукових досліджень. Проте, важливо пам'ятати про етичні аспекти та забезпечувати захист приватності користувачів.

## **1.2. Використання соціальних мереж у моделях прогнозування цін**

Використання соціальних мереж для прогнозування цін на товари є цінним інструментом, оскільки вони надають можливість отримувати відгуки від великої кількості користувачів щодо продуктів чи послуг. Аналіз цих відгуків може допомогти передбачити сприйняття товарів і їх попит, що, в свою чергу, дозволяє прогнозувати ціну на товар.

Для ефективного використання соціальних мереж у прогнозуванні цін необхідно врахувати усі важливі аспекти. По-перше, слід ретельно аналізувати дані, зібрані з соціальних мереж. Це включає структурування корисної інформації з коментарів та лайків, щоб виокремити справжню думку користувачів про товар серед великої кількості несуттєвих коментарів.

Подальший етап передбачає застосування методів аналізу, таких як статистичний аналіз і машинне навчання. Ці методи дозволяють виявити закономірності та тренди в даних, що сприяє точнішому прогнозуванню. Важливо також оцінити достовірність зібраної інформації, оскільки соціальні мережі не завжди є надійним джерелом. Існує ймовірність наявності проплачених коментарів, коли за грошову винагороду залишають позитивні відгуки про сумнівні товари. Також є сервіси, що автоматизують залишення коментарів за допомогою ботів, що значно ускладнює якісний аналіз.

Отже, для використання соціальних мереж у прогнозуванні цін на товари важливо вміти обирати і аналізувати інформацію з цих мереж з високою якістю. Це включає виділення релевантних даних, застосування відповідних методів аналізу та оцінку достовірності інформації. Тільки за таких умов соціальні мережі можуть стати корисним інструментом у прогнозуванні цін.

### **1.3 Характеристика інструментарію прогнозування цін**

Інструментарій прогнозування цін є важливим компонентом в економічному аналізі, який включає в себе різноманітні методи та моделі, що дозволяють передбачати майбутні зміни цін на товари та послуги. Основним завданням цих інструментів є забезпечення точності та надійності прогнозів, що в свою чергу дозволяє компаніям, інвесторам і державним установам приймати обґрунтовані рішення.

Для прогнозування цін застосовуються як кількісні, так і якісні методи. До кількісних методів належать часові ряди, регресійний аналіз, моделі авторегресії, методи ковзних середніх та інші статистичні техніки. Ці методи базуються на історичних даних і використовують математичні та статистичні інструменти для побудови моделей, що описують динаміку цін. Застосування часових рядів дозволяє виявити тренди та сезонні коливання, що є важливими для розуміння довгострокових змін цін. Регресійний аналіз дозволяє визначити залежність цін від різних факторів, таких як попит, пропозиція, витрати на виробництво, рівень інфляції та інші макроекономічні показники.

Якісні методи, з іншого боку, включають експертні оцінки, методи Делфі, аналіз сценаріїв та інші підходи, що спираються на досвід та інтуїцію фахівців. Ці методи особливо корисні в умовах високої невизначеності або коли немає достатньо історичних даних для застосування кількісних моделей. Метод Делфі, наприклад, передбачає опитування групи експертів у кілька раундів з метою досягнення консенсусу щодо прогнозованих цін.

Сучасні підходи до прогнозування цін також включають використання машинного навчання та штучного інтелекту. Ці методи дозволяють враховувати велику кількість змінних і виявляти складні взаємозв'язки між ними. Моделі на основі нейронних мереж, дерева рішень та інші алгоритми машинного навчання можуть автоматично адаптуватися до нових даних і покращувати точність прогнозів з часом. Використання великих даних і аналітичних платформ також забезпечує можливість обробки та аналізу величезних масивів інформації, що сприяє більш точному прогнозуванню.

Одним із ключових аспектів прогнозування цін є оцінка точності та надійності прогнозів. Для цього використовуються різні показники, такі як середня абсолютна похибка (MAPE), середньоквадратична похибка (RMSE) та інші метрики, що дозволяють порівняти прогнозовані значення з фактичними даними. Важливо також враховувати фактори ризику та невизначеності, що

можуть вплинути на точність прогнозів. Це може включати непередбачувані економічні події, зміни в політичному середовищі, природні катастрофи та інші фактори, що важко передбачити.

Отже, інструментарій прогнозування цін складається з різноманітних методів та моделей, які мають свої переваги та обмеження. Вибір конкретного інструменту залежить від доступності даних, специфіки ринку, обраних часових горизонтів прогнозування та інших факторів. Комбінування кількісних і якісних підходів, а також використання сучасних технологій машинного навчання дозволяє досягти більшої точності та надійності прогнозів, що є ключовим для успішного прийняття рішень в умовах мінливого економічного середовища.

#### **1.4. Аналіз методів прогнозування цін на товари за допомогою метрик соціальних мереж**

Аналіз методів прогнозування цін на товари за допомогою метрик соціальних мереж є актуальною і важливою темою у сучасній економіці та бізнес-аналітиці. Соціальні мережі вже давно є частиною нашого життя, і їхній вплив на поведінку споживачів, а також на формування ринкових трендів, є незаперечним. Метрики соціальних мереж, такі як лайки, коментарі, репости та кількість підписників, можуть служити цінними індикаторами для прогнозування змін у попиті на товари та, відповідно, цін на них.

Використання даних з соціальних мереж для прогнозування цін на товари базується на аналізі великого обсягу інформації, яка генерується користувачами щодня. Одним із підходів є застосування методів машинного навчання, що дозволяють обробляти та аналізувати ці дані. Такі методи включають класифікацію, регресію та кластеризацію, які можуть бути використані для виявлення закономірностей та тенденцій у поведінці споживачів. Наприклад, різке зростання кількості згадувань певного товару в соціальних мережах може

сигналізувати про підвищення інтересу до цього товару, що, в свою чергу, може призвести до підвищення його ціни.

Інший важливий аспект аналізу метрик соціальних мереж полягає у врахуванні контексту згадувань товару. Позитивні та негативні відгуки, емоційне забарвлення коментарів можуть значно впливати на сприйняття товару споживачами. Аналіз тональності текстів, що публікуються у соціальних мережах, дозволяє більш точно прогнозувати зміни у попиті та відповідні коливання цін. Для цього використовуються алгоритми обробки природної мови (NLP), які допомагають автоматично розпізнавати емоції та тональність тексту.

Важливою перевагою використання метрик соціальних мереж є їхня оперативність. Дані з соціальних мереж можна збирати та аналізувати в режимі реального часу, що дозволяє миттєво реагувати на зміни у ринковій ситуації. Це особливо цінно в умовах швидкоплинного ринку, де своєчасне прийняття рішень може забезпечити конкурентну перевагу.

Однак, незважаючи на численні переваги, використання даних соціальних мереж для прогнозування цін на товари має і свої обмеження. Одним із головних викликів є шумність даних: велика кількість несуттєвої або неправдивої інформації може ускладнювати аналіз. Крім того, соціальні мережі охоплюють не всі демографічні групи, що може призвести до викривлення результатів прогнозування. Необхідно також враховувати питання конфіденційності та етичні аспекти збору та використання даних користувачів.

Таким чином, аналіз метрик соціальних мереж для прогнозування цін на товари представляє собою перспективний напрямок, що поєднує сучасні технології з традиційними методами економічного аналізу. Він дозволяє отримувати цінну інформацію про ринкові тенденції та споживчі настрої, сприяючи прийняттю більш обґрунтованих управлінських рішень. Водночас, для досягнення максимальної ефективності необхідно враховувати специфіку даних соціальних мереж та забезпечувати їх коректну інтерпретацію.

## Висновки за розділом 1

У розділі, присвяченому теоретичним засадам, моделям та методам прогнозування цін на товари з використанням метрик соціальних мереж, підкреслено важливість інтеграції сучасних технологій аналізу великих даних та машинного навчання у процесі економічного прогнозування. Зокрема, розглядаються різні підходи до збору та обробки даних з соціальних мереж, які можуть слугувати ефективними індикаторами зміни цін на товари. Акцентується увага на тому, що аналіз текстової інформації, зокрема настроїв споживачів, в соціальних мережах, може надавати цінні підказки щодо майбутніх змін попиту і, відповідно, цін.

Описані математичні моделі, які використовуються для прогнозування, включають регресійний аналіз, нейронні мережі, методи машинного навчання та інші інструменти обробки природної мови (NLP). Показано, як різні типи моделей можуть бути поєднані для підвищення точності прогнозування. Наприклад, регресійний аналіз може бути застосований для виявлення основних тенденцій, в той час як нейронні мережі можуть забезпечувати більш детальний аналіз складних нелінійних взаємозв'язків у даних.

Методи збору даних з соціальних мереж включають скрапінг, аналіз контенту та обробку великих даних. Зібрана інформація піддається фільтрації, очищенню та нормалізації для подальшого аналізу. Це забезпечує підвищення якості даних, що використовуються в моделях прогнозування. Крім того, використання метрик соціальних мереж дозволяє враховувати не лише кількісні показники, наприклад частоту згадок певного товару, але й якісні характеристики, такі як тональність коментарів і відгуків.

Загалом, підкреслено, що інтеграція даних з соціальних мереж у процесі прогнозування цін на товари відкриває нові можливості для більш точного і

оперативного аналізу ринкових тенденцій. Це дозволяє компаніям краще адаптуватися до змін на ринку, реагувати на зміни в споживчих настроях і, відповідно, оптимізувати свої стратегії ціноутворення. Водночас, відзначено важливість подальших досліджень і розвитку методологій, що дозволять ще більш ефективно використовувати потенціал соціальних мереж у прогнозуванні економічних показників.

## РОЗДІЛ 2

### МОДЕЛЮВАННЯ СТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОГНОЗУВАННЯ ЦІН

#### 2.1 Засоби агрегації неструктурованих даних

Для ефективного прогнозування ціни на товар, базуючись на метриках соціальних мереж, необхідно застосовувати сучасні засоби агрегації неструктурованих даних. Такий підхід дозволяє отримати цілісне уявлення про динаміку попиту, поведінку споживачів і загальні тенденції на ринку. Одним із основних методів агрегації є застосування технологій обробки природної мови (NLP), які дозволяють перетворювати текстову інформацію з соціальних мереж у структуровані дані. Наприклад, використання алгоритмів машинного навчання для аналізу тональності текстів, виявлення ключових тем і трендів, а також класифікації відгуків та коментарів споживачів.

Важливим аспектом є також збір і обробка великих обсягів даних у реальному часі. Для цього застосовуються платформи великих даних, такі як Hadoop або Spark, які дозволяють ефективно управляти потоком інформації, проводити її фільтрацію та очищення. Зокрема, такі платформи можуть інтегрувати дані з різних джерел, таких як: соціальні мережі, блоги, форуми та новинні сайти, що сприяє формуванню більш точної картини ринкових умов.

Додатково використовуються технології web scraping для автоматизованого збору даних з веб-сторінок. Це дозволяє отримувати актуальну інформацію про зміни у вподобаннях споживачів, нові тренди та реакції на певні події. Однак, процес web scraping пов'язаний з певними технічними і юридичними складнощами, такими як необхідність обходу захисту від ботів і дотримання правил конфіденційності.

Не менш важливою є аналітична обробка зібраних даних. Тут використовуються методи статистичного аналізу і візуалізації даних для виявлення закономірностей і трендів. Інструменти, такі як Tableau або Power BI, дозволяють створювати наочні графіки та діаграми, що полегшують інтерпретацію результатів аналізу. Це допомагає приймати обґрунтовані рішення щодо ціноутворення товарів, базуючись на реальних даних, а не лише на інтуїції або припущеннях.

Варто зазначити, що ефективна агрегація неструктурованих даних потребує міждисциплінарного підходу, який об'єднує знання в галузі інформатики, статистики, маркетингу та економіки. Такий комплексний підхід дозволяє отримати більш точні прогнози та краще розуміти поведінку ринку. Сучасні інструменти та методи агрегації даних відкривають нові можливості для бізнесу, дозволяючи швидко реагувати на зміни ринкових умов і забезпечувати конкурентоспроможність товарів.

На завершення слід підкреслити, що процес агрегації неструктурованих даних є складним і багатогранним. Він включає збір, обробку, аналіз і інтерпретацію великих обсягів інформації, що вимагає високого рівня технічних знань і ресурсів. Проте, завдяки правильному використанню сучасних технологій, можна досягти значних результатів у прогнозуванні цін на товари та підвищенні ефективності бізнес-стратегій.

## **2.2 Обґрунтування вибору середовища розробки програмної системи**

При виборі програмного забезпечення для обробки та аналізу метрик соціальних мереж насамперед необхідно визначити функціональні характеристики.

Програмне забезпечення має збирати дані з різних джерел. Джерелами можуть виступати: Facebook, Twitter, Instagram. Таким чином, воно має

взаємодіяти з API відповідних платформ для отримання доступу до даних. Після отримання даних програмне забезпечення має можливість для сортування даних та їх фільтрації. Це повинно бути з метою покращення результатів аналізу.

Після збирання інформації потрібно її проаналізувати. Програмне забезпечення має виявляти тенденції в думці користувачів щодо бренду товару, визначати ключові фактори що впливають на ціноутворення та ставлення користувачів.

Після збирання та аналізу даних платформа повинна ці дані візуалізувати. Це робиться для того, щоб виявити залежності та інтерпретувати результати аналізу. Для візуалізації даних використовуються різні графіки, діаграми.

Після визначення функціональних вимог до програмного забезпечення необхідно також визначити вимоги до його інтерфейсу.

Інтерфейс має важливе значення для функціонування програми. Він дозволяє користувачам швидко знаходити потрібні функції та виконувати свої завдання. Інтерфейс повинен бути інтуїтивно зрозумілим користувачам. При оцінці інтерфейсу застосовуються характеристики, такі як: доступність функціоналу, послідовність кроків для виконання завдань, наявність підказок.

Також важливим аспектом інтерфейсу є його логіка. Це необхідно, щоб користувачі не витрачали час на пошук необхідної функції в структурі програми.

Останнім аспектом при виборі програмного забезпечення є його інтеграція з іншими аналітичними інструментами та соціальними мережами. Для інтеграції використовується API, що можуть надавати доступ до соціальних мереж для взаємодії з сторонніми програмами. API може надавати доступ до публічних постів, коментарів, аудиторію. Програмне забезпечення буде використовувати API для автоматизованого збору інформації та її аналізу з соціальних мереж.

Іншим підходом для інтеграції може бути використання модулів чи плагінів. Вони можуть бути розроблені спеціально для інтеграції з соціальними мережами.

Для розробки програмного забезпечення, виходячи з вимог вище, було обрано мову програмування Python.

Python – мова програмування високого рівня, яка має простий та зрозумілий синтаксис, а також багато бібліотек для обробки даних[7]. Використання цих бібліотек допоможе у написанні програмного забезпечення для аналізу метрик соціальних мереж. Крім цього Python також має бібліотеки для візуалізації даних, що надасть можливість наочно, у вигляді графіків, побачити результати аналізу соціальних мереж та зробити висновки.

Для інтеграції з соціальними мережами буде застосовано різні API. Для інтеграції з твітером в Python передбачена бібліотека Твеєру[7]. Вона надає можливість інтеграції з списками друзів, даними щодо постів, взаємодій. Також можна застосовувати Facebook та Instagram Graph API, які дозволять інтегрувати дані з цих соціальних мереж до програмного забезпечення для їх подальшого аналізу.

Для розробки програмного забезпечення буде використано середовище PyCharm (рис.2.1).

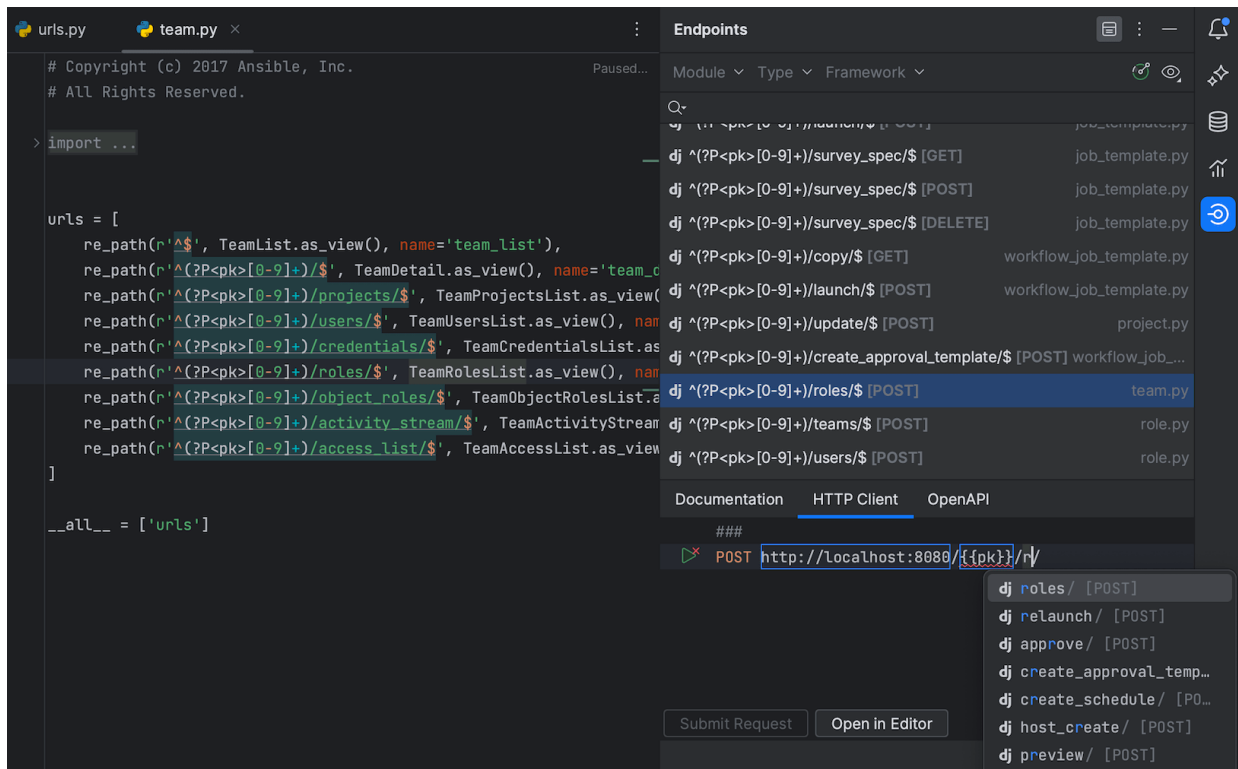


Рисунок 2.1 – Вікно середовища PyCharm

Це інтегроване програмне забезпечення для розробки на мові програмування Python. Воно має кілька основних переваг:

1. Розширена підтримка Python.
2. Редагування коду.
3. Візуальне налаштування.
4. Підтримка інших мов програмування.
5. Підтримка фреймворків.
6. Спільні проєкти.

Програмне забезпечення для аналізу даних з соціальних мереж є важливим та необхідним етапом для прогнозування цін. Воно дозволяє збирати, аналізувати та візуалізувати дані, після чого робити прогнози щодо зміни ціни.

### 2.3 Побудова UML-діаграм предметної області

Використання UML-діаграм для моделювання предметної області прогнозування дозволяє систематизувати та візуалізувати процеси, що відбуваються в рамках розробки програмного забезпечення. Це, в свою чергу, сприяє кращому усвідомленню вимог до системи, оптимізації ресурсів та мінімізації можливих ризиків.

Побудова UML-діаграм предметної області є важливим етапом у створенні системи прогнозування цін на товари. За допомогою діаграм класів, об'єктів, діяльності та інших типів UML-діаграм можливо наочно відобразити структуру даних, взаємозв'язки між елементами системи та логіку її функціонування. Такий підхід допомагає розробникам, аналітикам та іншим учасникам проекту краще розуміти складні аспекти предметної області та координувати свої дії. Крім того, UML-діаграми сприяють створенню більш надійної та гнучкої системи, що може адаптуватися до змін вимог ринку та нових викликів. Таким чином, побудова UML-діаграм предметної області є необхідною складовою успішного проектування систем прогнозування цін на товари на основі метрик соціальних мереж.

Діаграма варіантів використання моделі представлена на рисунку 2.2.

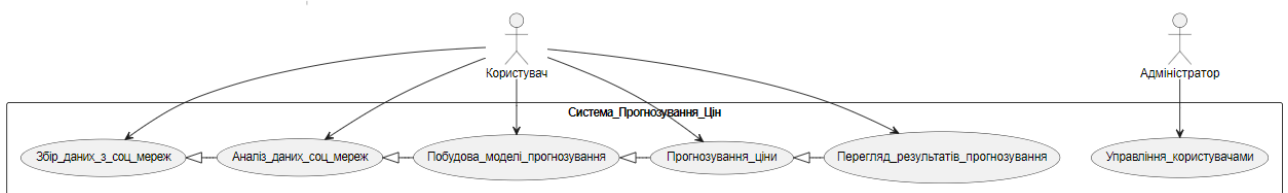


Рисунок 2.2 – Діаграма варіантів використання предметної області

Створена діаграма варіантів використання відображає основні функціональні можливості системи прогнозування цін на товари, заснованої на метриках соціальних мереж. Основними акторів у системі є Користувач та Адміністратор, кожен з яких має свої особливі ролі та функціональні обов'язки.

Користувач взаємодіє з системою через п'ять основних варіантів використання: збір даних з соціальних мереж, аналіз цих даних, побудова моделі прогнозування, здійснення прогнозування ціни та перегляд результатів прогнозування. Ці варіанти використання пов'язані між собою, що дозволяє зрозуміти послідовність виконання дій та їх взаємозалежність.

Адміністратор системи відповідає за управління користувачами, що включає в себе створення нових користувачів, налаштування прав доступу та інші адміністративні функції. Варіанти використання "Збір даних з соц мереж" та "Аналіз даних соц мереж" відображають початкові етапи роботи з системою, де здійснюється отримання та попередня обробка інформації. Далі, на основі проаналізованих даних, проводиться "Побудова моделі прогнозування", що є ключовим етапом у процесі створення прогнозу. Після цього здійснюється безпосереднє "Прогнозування ціни", результат якого користувач може переглянути через варіант використання "Перегляд результатів прогнозування". Усі ці варіанти використання інтегровані в єдину систему, що забезпечує зручність та ефективність роботи користувачів із системою прогнозування цін на товари.

Діаграма послідовності представлена на рисунку 2.3.

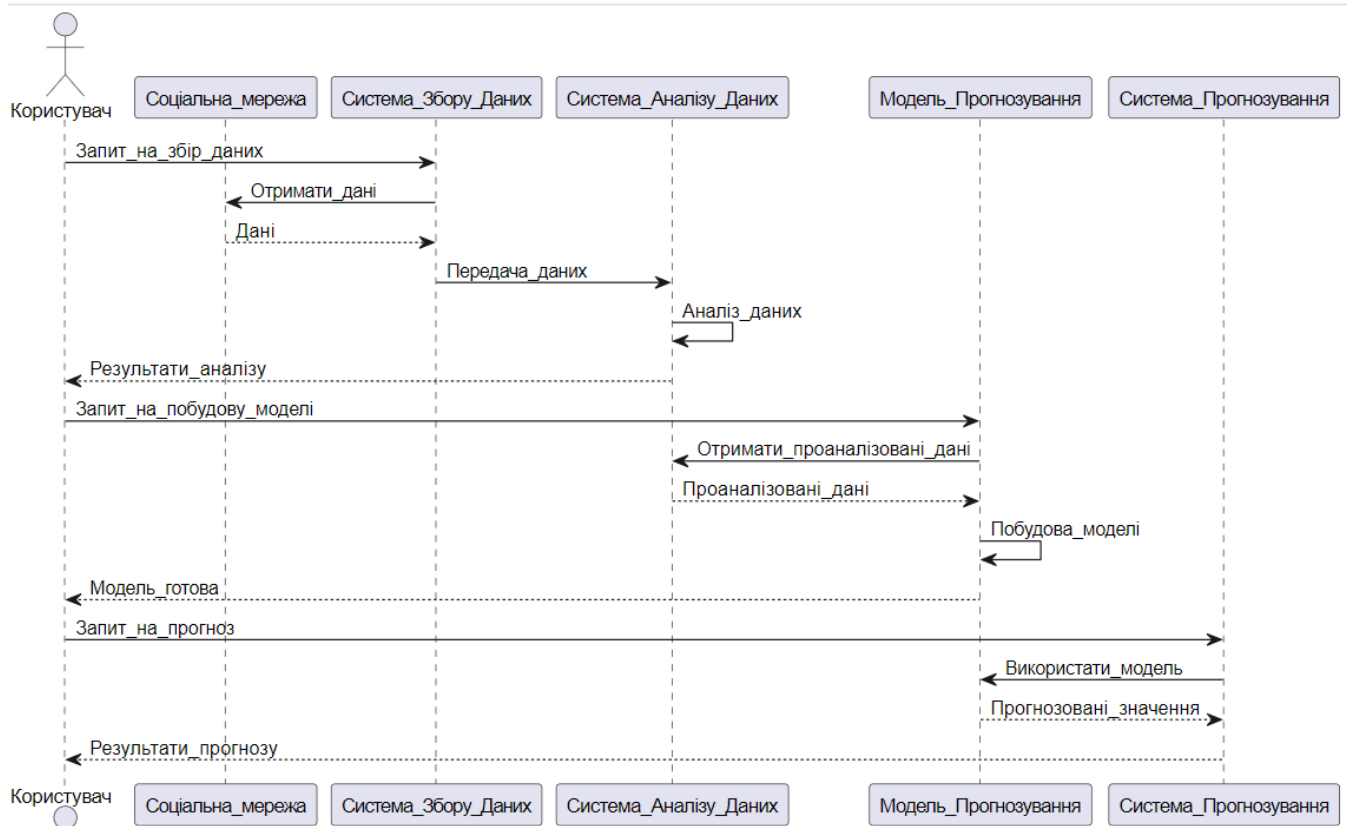


Рисунок 2.3 – Діаграма послідовності предметної області

Створена діаграма послідовності відображає процес взаємодії користувача з системою прогнозування цін на товари, яка використовує дані з соціальних мереж. Користувач ініціює взаємодію із системою через запит на збір даних, що передається до системи збору даних. Система збору даних, у свою чергу, надсилає запит до соціальної мережі для отримання необхідної інформації. Отримані дані передаються назад до системи збору даних, а потім ці дані пересилаються до системи аналізу даних, де здійснюється їх детальний аналіз. Результати аналізу повертаються користувачу, що дозволяє йому зробити висновки про стан ринку та поведінку споживачів.

Після отримання результатів аналізу користувач надсилає запит на побудову моделі прогнозування, використовуючи проаналізовані дані. Цей запит передається до відповідного компонента системи, який здійснює побудову

моделі на основі отриманих даних. Після завершення побудови моделі користувач отримує підтвердження про готовність моделі до використання. Наступним кроком є запит користувача на прогнозування ціни товару. Цей запит надсилається до системи прогнозування, яка використовує побудовану модель для обчислення прогнозованих значень. Результати прогнозування повертаються користувачу, дозволяючи йому отримати точні прогнози щодо цін на товари на основі метрик соціальних мереж. Цей процес забезпечує комплексний підхід до аналізу та прогнозування, використовуючи сучасні технології для прийняття обґрунтованих рішень.

Діаграма класів системи представлена на рисунку 2.4.

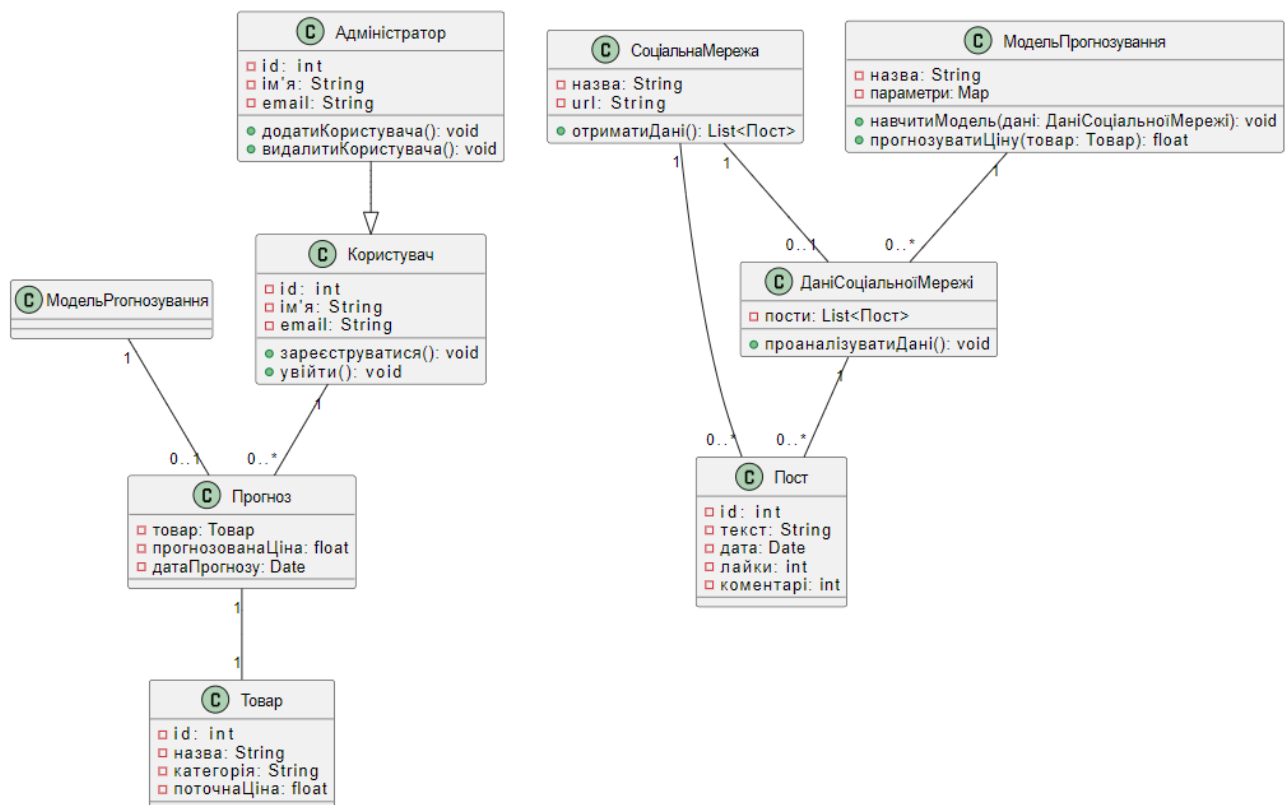


Рисунок 2.4 – Діаграма класів предметної області

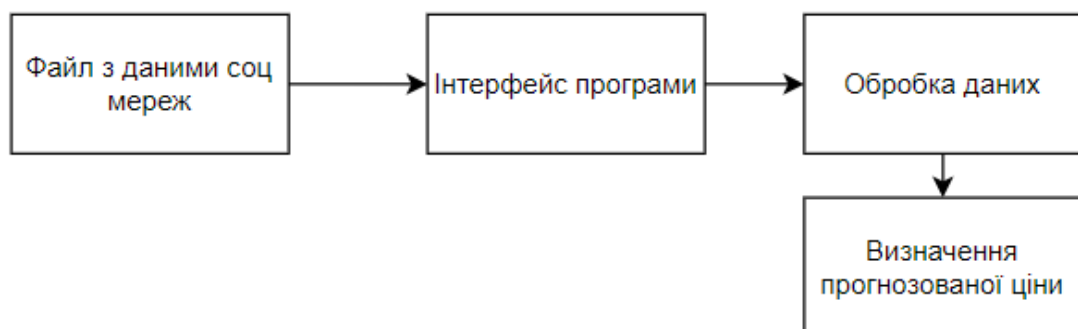
Дивлячись на цю діаграму, маємо розгорнутий зразок того, як виглядає вся система прогнозування цін на товари на основі даних з соціальних мереж. У ній

можна помітити кілька основних класів, які представляють різні аспекти системи. Спершу - це класи, пов'язані з користувачами: є клас "Користувач" і "Адміністратор", які мають свої особливості та можливості. Потім - це класи, які відповідають за дані з соціальних мереж: "СоціальнаМережа", "Пост" та "ДаніСоціальноїМережі". Ці класи відображають структуру та характеристики даних, які система збирає та аналізує. Далі – це класи, пов'язані з прогнозуванням: "МодельПрогнозування", "Товар" і "Прогноз". Ці класи відображають логіку та процес прогнозування цін на товари.

Усі ці класи взаємодіють між собою через методи та атрибути, які показані на діаграмі. Наприклад, користувачі можуть реєструватися та увійти в систему, адміністратор може додавати та видаляти користувачів, а соціальна мережа може надавати дані, які потім аналізуються та використовуються для прогнозування цін. Кожен клас має свої унікальні характеристики та функціональні можливості, а взаємодія між ними дозволяє системі працювати як єдине ціле. Таким чином, діаграма класів предметної області прогнозування цін на товари на основі метрик соціальних мереж надає важливий огляд структури та функціональності системи, що допомагає розуміти, як вона працює та які можливості вона надає.

## 2.4 Розробка архітектури програмної системи

Архітектура програмної системи представлена на рисунку 2.5.



### Рисунок 2.5 – Архітектура програмного забезпечення для прогнозування ціни на товар на основі метрик соціальних мереж

Подана схема відображає процес обробки даних соціальних мереж для визначення прогнозованої ціни на товар. Початковим етапом є наявність файлу з даними соціальних мереж, що містить неструктуровану інформацію, отриману з різних платформ. Ці дані можуть включати тексти постів, коментарі, лайки, шери та інші показники активності користувачів, які відображають їхні вподобання та поведінку.

Наступним етапом є інтерфейс програми, через який здійснюється завантаження цього файлу. Інтерфейс виступає посередником між сирими даними і алгоритмами обробки, забезпечуючи їхню коректну передачу до наступного блоку. Інтерфейс програми може мати різні форми, включаючи графічні або командні оболонки, які дозволяють користувачеві легко завантажувати файли та налаштовувати параметри обробки.

Після завантаження даних у систему, відбувається їхня обробка. Цей процес включає кілька ключових етапів: фільтрація та очищення даних, виділення ключових характеристик, аналіз тональності, класифікація та інші методи обробки природної мови. Мета обробки полягає у перетворенні неструктурованих даних у структуровану форму, що придатна для подальшого аналізу. Застосовуються алгоритми машинного навчання, статистичні методи та інші інструменти аналітики для виявлення трендів і закономірностей у зібраних даних.

Завершальним етапом є визначення прогнозованої ціни на основі оброблених даних. На цьому етапі використовуються моделі прогнозування, які враховують виявлені тренди, кореляції та інші фактори, що впливають на цінову динаміку. Результати прогнозування можуть надавати цінну інформацію для прийняття стратегічних рішень у бізнесі, дозволяючи краще розуміти ринкові

умови і адаптувати цінову політику відповідно до очікувань споживачів та поточних трендів.

Таким чином, дана схема ілюструє послідовність кроків, що необхідні для ефективного використання даних соціальних мереж у прогнозуванні цін. Кожен етап має свої специфічні задачі і використовує різноманітні технології та методи для досягнення кінцевої мети - точного і обґрунтованого прогнозу цін на товари.

## **Висновки за розділом 2**

Після уважного розгляду різних варіантів програмного забезпечення для розробки моделі прогнозування цін на товари, було вирішено обрати Python у поєднанні з PyChart як найбільш відповідну альтернативу для наших потреб. Аналізуючи його архітектуру, ми звернули увагу на можливість створення файлу, який слугуватиме основою програми, що значно спрощує робочий процес та полегшує обробку даних. Крім того, враховуючи потреби в навчанні моделі та наданні прогнозованої ціни, було зазначено, що Python у поєднанні з PyChart надає необхідні інструменти та функціонал для цих завдань.

У процесі розробки такої моделі було використано UML-діаграми предметної області, які включали в себе діаграми варіантів використання, послідовності та класів. Ці діаграми допомогли краще зрозуміти та візуалізувати процеси, що відбуваються в системі прогнозування цін на товари. Наприклад, діаграми варіантів використання дозволили уявити, які конкретно дії може виконати користувач або адміністратор в системі. Діаграми послідовності допомогли зрозуміти взаємодію між різними компонентами системи та послідовність виконання операцій. А діаграми класів дозволили уявити структуру даних та взаємозв'язки між різними класами програми.

Такий підхід до аналізу архітектури програмного забезпечення та розробки UML-діаграм допоміг у створенні чіткого та ефективного інструменту для прогнозування цін на товари на основі метрик соціальних мереж.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРОГНОЗУВАННЯ ЦІН

#### **3.1 Програмна реалізація методу прогнозування цін на товари на основі метрик соціальних мереж**

Програмна реалізація методу прогнозування цін на товари на основі метрик соціальних мереж включає кілька важливих етапів, які забезпечують повноцінне функціонування системи та отримання точних прогнозів. Основними компонентами системи є збір даних, їх обробка, навчання моделей, і, власне, прогнозування цін. Впровадження цих компонентів реалізовано за допомогою мови програмування Python та бібліотек PyQt5, pandas, numpy, sqlite3, sklearn, та інших. Розглянемо основні аспекти даної реалізації детальніше.

#### **Збір та зберігання даних**

Збір даних є першочерговим кроком у процесі прогнозування. Програма дозволяє користувачам додавати інформацію про товари та пости з соціальних мереж. Дані про товари включають назву, ціну та категорію. Інформація про пости містить дані про платформу, кількість вподобань, коментарів, аудиторію та кількість днів з моменту публікації. Для зберігання цієї інформації використовується SQLite база даних, що забезпечує легкість у роботі з даними та їх надійне зберігання.

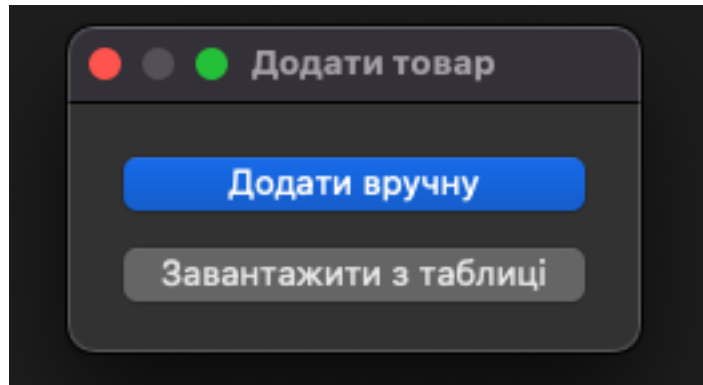


Рисунок 3.1 – приклад додавання товару

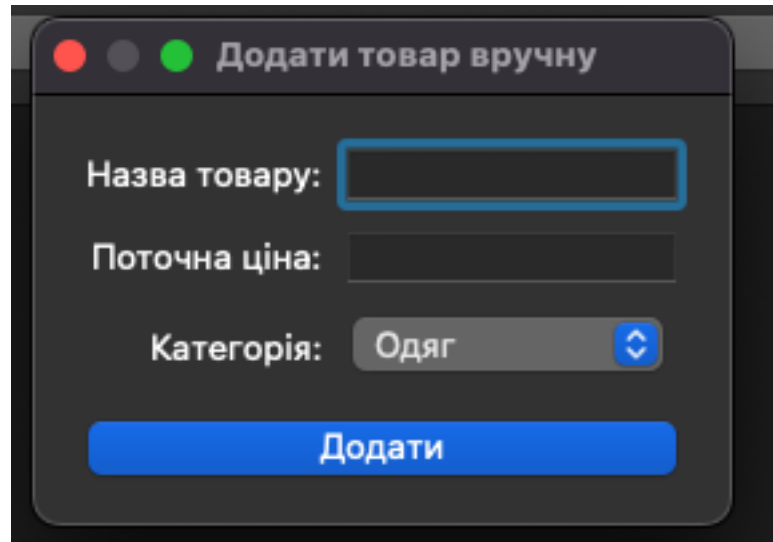


Рисунок 3.2 – приклад інтерфейсу ручного додавання товару

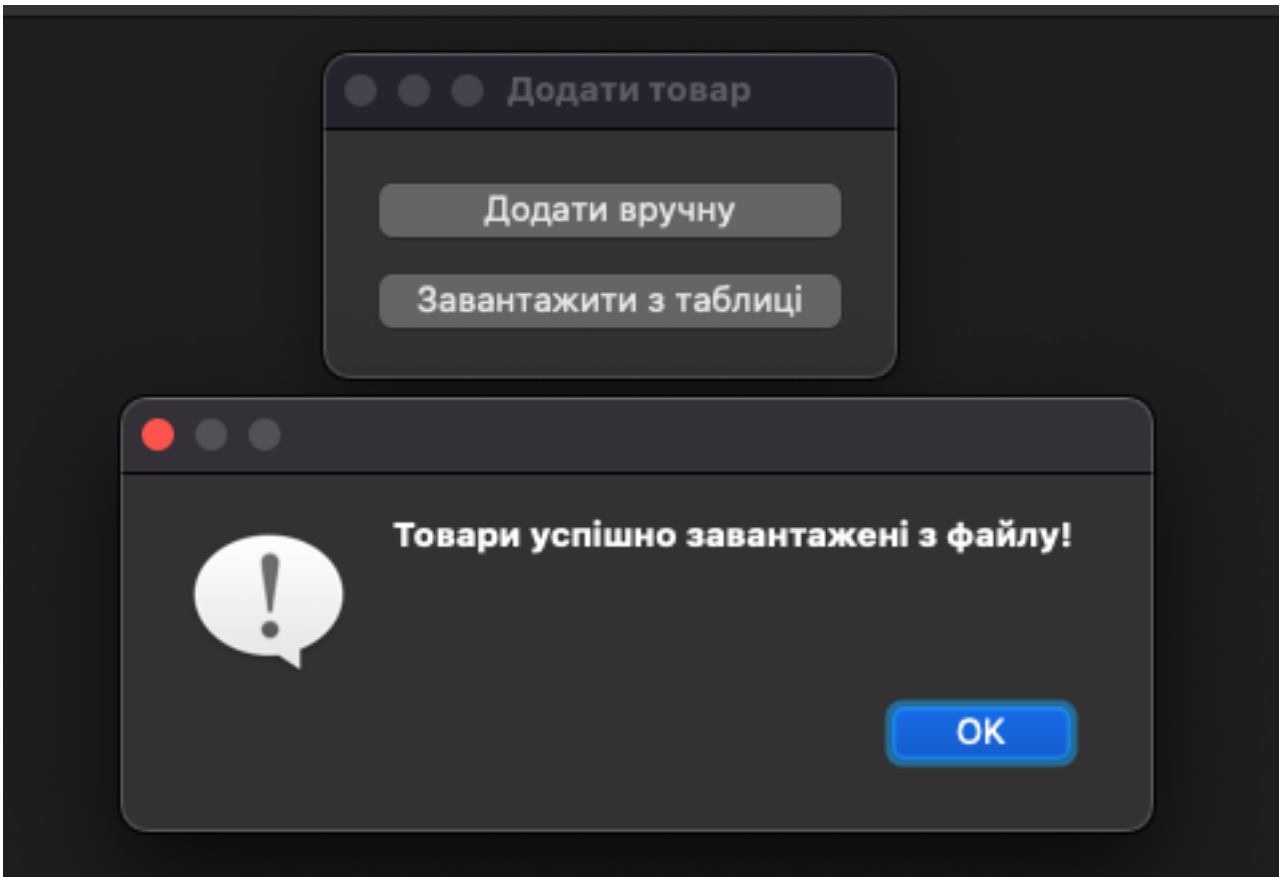


Рисунок 3.3 – приклад успішного завантаження товарів з таблиці

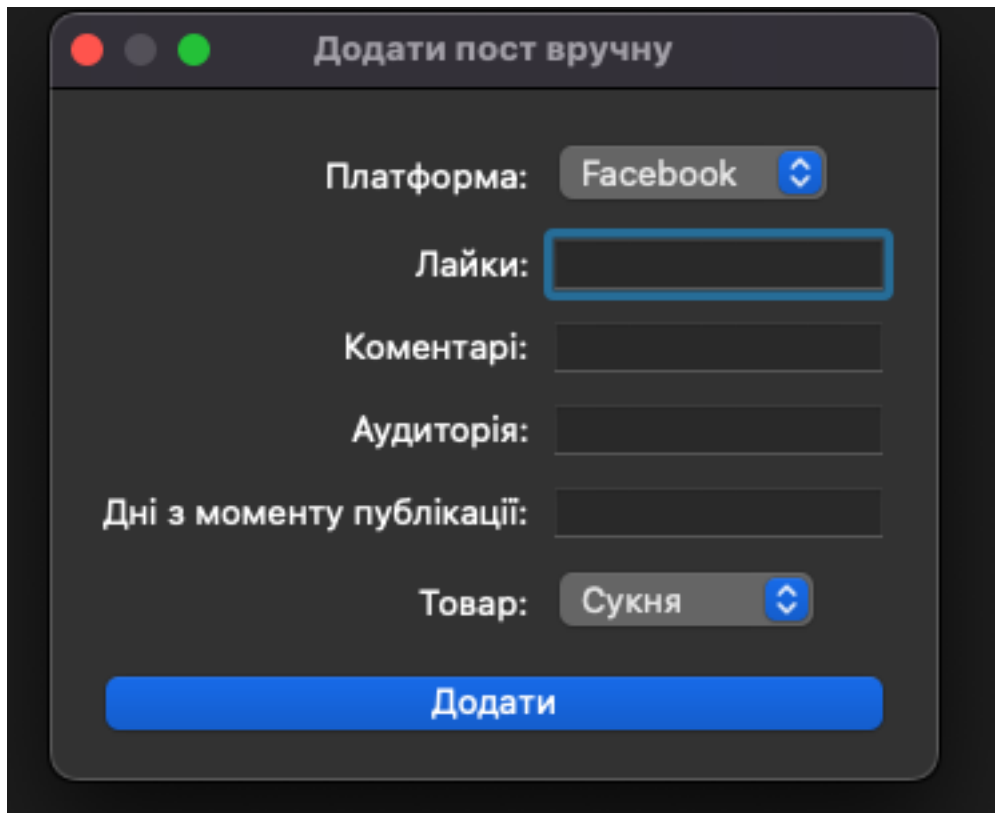


Рисунок 3.4 – приклад інтерфейсу для додавання посту

### Інтерфейс користувача

Графічний інтерфейс користувача розроблено за допомогою бібліотеки PyQt5, що надає зручні засоби для взаємодії користувача з системою. Основні функції інтерфейсу включають додавання нових товарів та постів, перегляд існуючих записів, оновлення та видалення записів, а також запуск процесів навчання моделі та прогнозування цін.

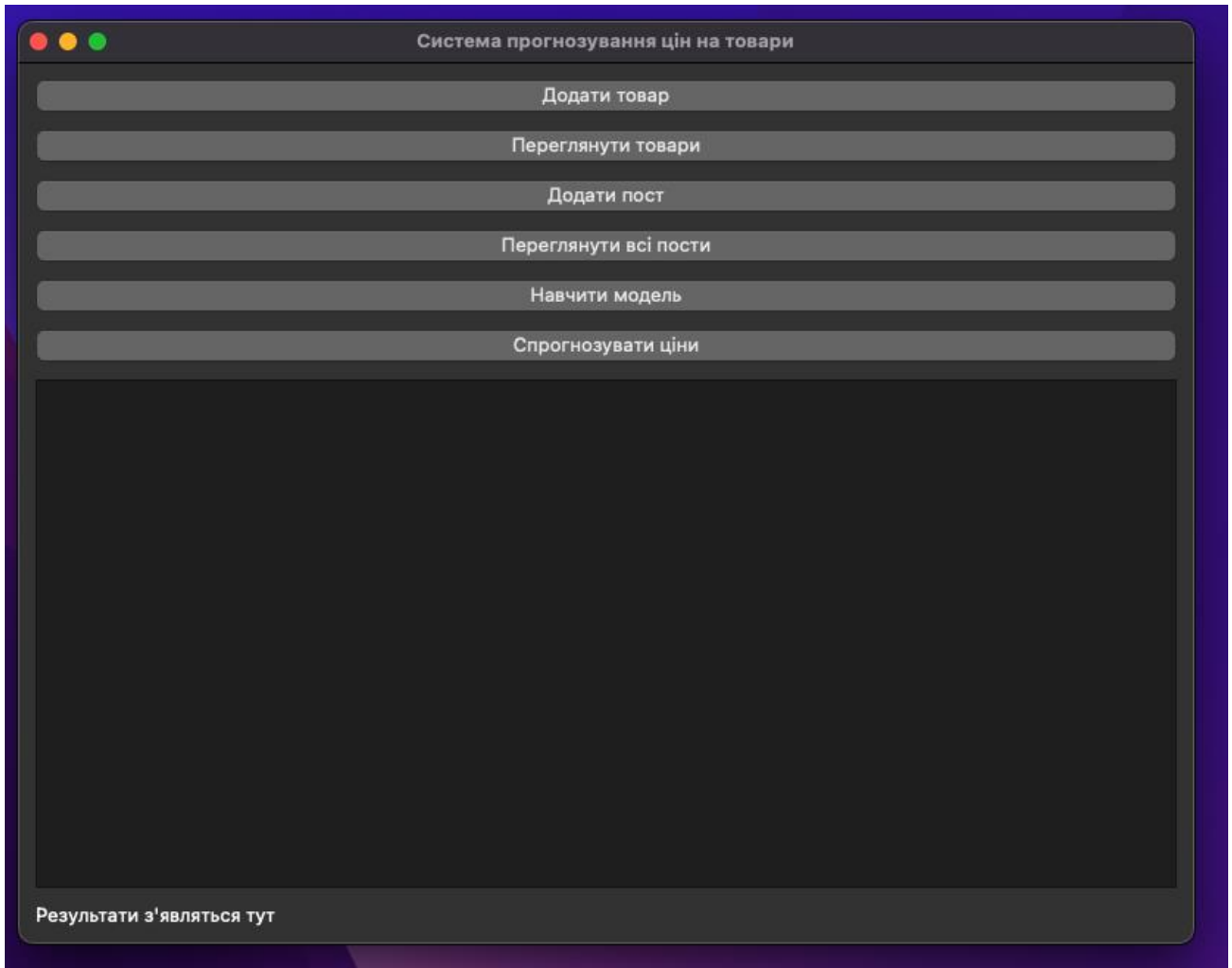


Рисунок 3.5 – головний інтерфейс програми

### **Обробка та аналіз даних**

Для обробки даних використовується бібліотека `pandas`, яка дозволяє ефективно маніпулювати табличними даними та здійснювати їх аналіз. Основні етапи обробки включають фільтрацію та очищення даних, нормалізацію та підготовку до подальшого використання в моделях машинного навчання.

	ID	Платформа	Лайки	Коментарі	Аудиторія	з моменту публі	Товар
1	2	Facebook	1200	45	10000	7	Сукня
2	3	Instagram	8000	150	50000	5	Телефон
3	4	Twitter	500	30	2000	10	Лялька
4	5	Instagram	300	10	1500	3	Книга
5	6	Facebook	2000	70	25000	2	Ноутбук
6	7	Facebook	1200	45	10000	7	Сукня
7	8	Instagram	8000	150	50000	5	Телефон
8	9	Twitter	500	30	2000	10	Лялька
9	10	Instagram	300	10	1500	3	Книга
10	11	Facebook	2000	70	25000	2	Ноутбук

Оновити пост

Видалити пост

Рисунок 3.6 - Скріншот інтерфейсу для перегляду та редагування постів

### Навчання моделей

Навчання моделей здійснюється з використанням різних алгоритмів машинного навчання, таких як лінійна регресія, дерево рішень та Random Forest. Для реалізації цих алгоритмів використовується бібліотека sklearn, яка надає необхідний інструментарій для побудови та оцінки моделей. Процес навчання включає вибір параметрів моделі, підготовку тренувального набору даних та виконання перехресної перевірки для оцінки якості моделі.

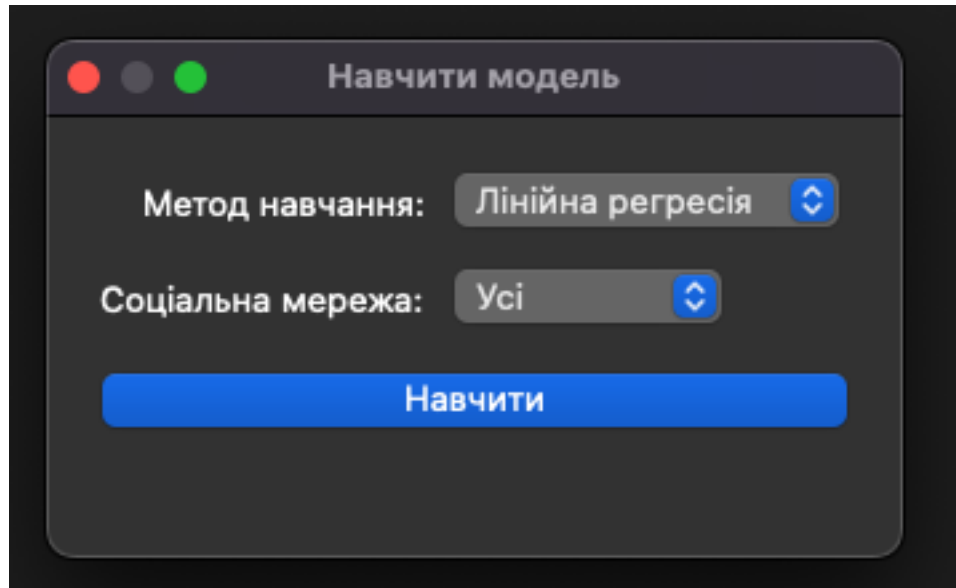
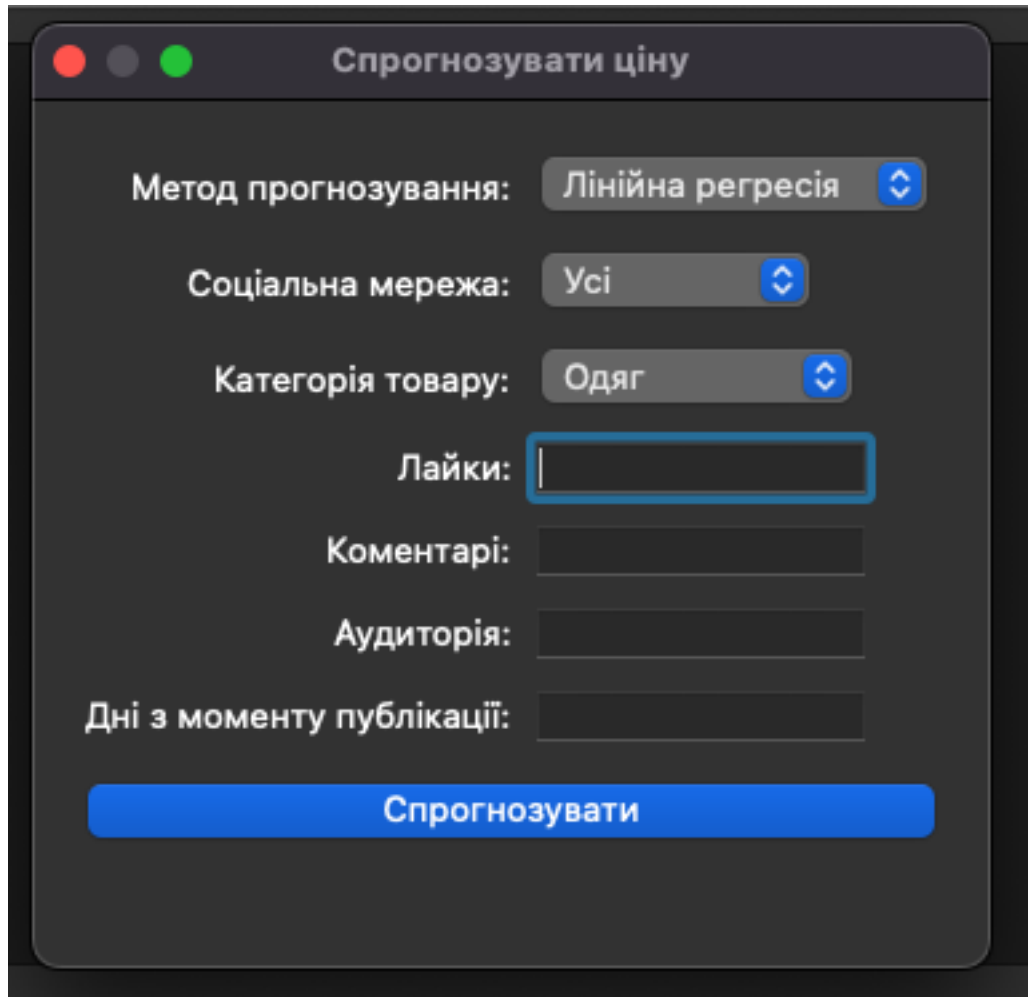


Рисунок 3.7 - Скріншот інтерфейсу для навчання моделей

### **Прогнозування цін**

Після навчання моделі користувач може здійснити прогнозування цін на основі введених параметрів. Для цього використовується відповідна модель, збережена у файлі формату .pkl, що дозволяє швидко завантажувати модель та використовувати її для прогнозування. Результат прогнозування виводиться на екран у вигляді прогнозованої ціни та середньої ціни в категорії.



Спрогнозувати ціну

Метод прогнозування: Лінійна регресія

Соціальна мережа: Усі

Категорія товару: Одяг

Лайки:

Коментарі:

Аудиторія:

Дні з моменту публікації:

Спрогнозувати

Рисунок 3.8 - Скріншот інтерфейсу для прогнозування цін

### **3.2 Експериментальне дослідження прикладного аспекту ціноутворення за допомогою системи прогнозування**

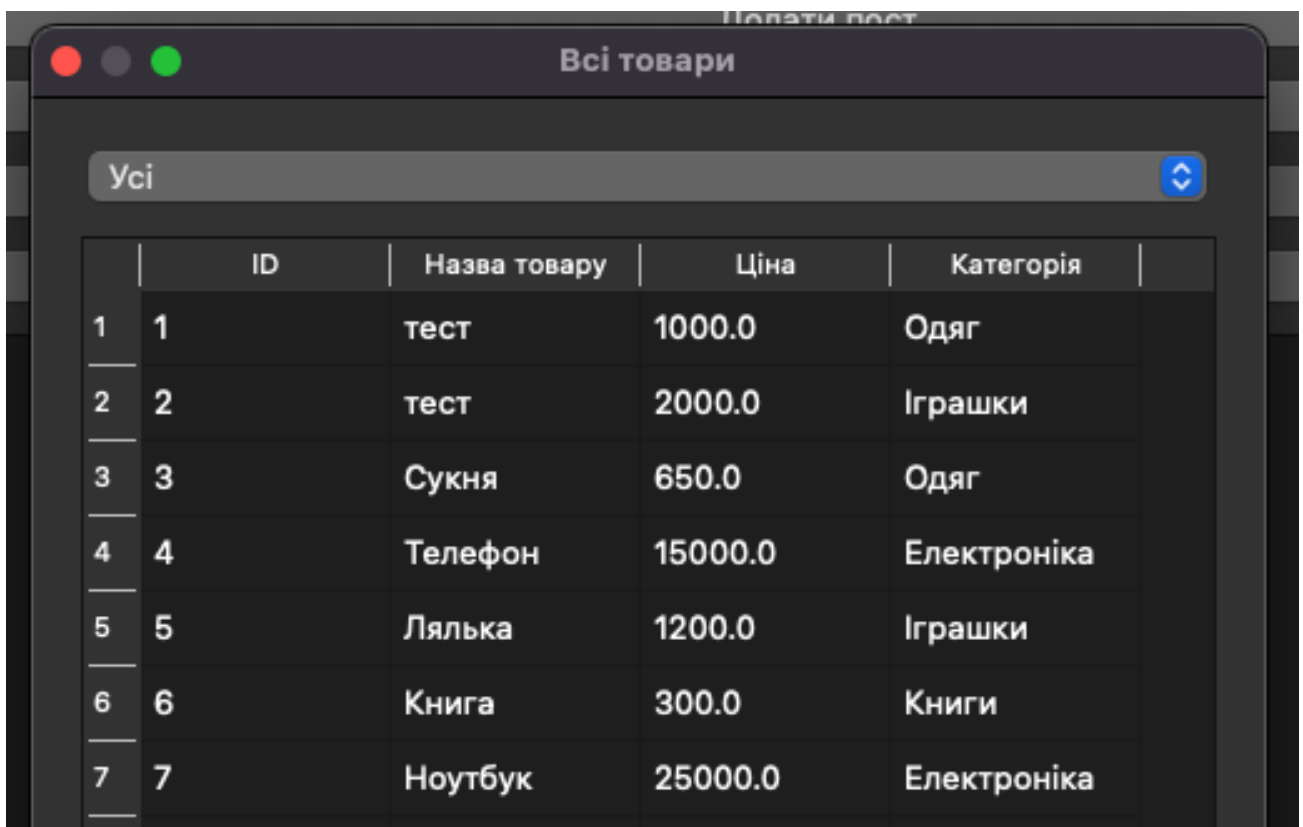
Експериментальне дослідження прикладного аспекту ціноутворення передбачає оцінку точності та ефективності розробленої моделі прогнозування цін на основі даних соціальних мереж. Метою цього дослідження є визначення практичної цінності моделі для бізнесу та її здатності забезпечувати точні прогнози в реальних умовах.

#### **Підготовка даних для експерименту**

Першим етапом експерименту було збирання даних з кількох соціальних мереж, включаючи метрики, такі як кількість вподобань, коментарів, розмір аудиторії, та кількість днів з моменту публікації. Дані були структуровані та внесені до бази даних програми.

### **Вибір товарів для тестування**

Для тестування було обрано кілька товарів з різних категорій, включаючи одяг, електроніку, іграшки та книги. Кожен товар був асоційований з відповідними постами в соціальних мережах, що містили необхідну інформацію для прогнозування цін.



	ID	Назва товару	Ціна	Категорія
1	1	тест	1000.0	Одяг
2	2	тест	2000.0	Іграшки
3	3	Сукня	650.0	Одяг
4	4	Телефон	15000.0	Електроніка
5	5	Лялька	1200.0	Іграшки
6	6	Книга	300.0	Книги
7	7	Ноутбук	25000.0	Електроніка

Рисунок 3.9 - Скріншот інтерфейсу для перегляду товарів та їх характеристик

### **Навчання та валідація моделей**

На основі зібраних даних було проведено навчання моделей з використанням алгоритмів лінійної регресії, дерева рішень та Random Forest.

Для оцінки якості моделей використовувалася перехресна перевірка (cross-validation), що дозволило визначити середню похибку моделі та її стійкість до зміни вхідних даних.

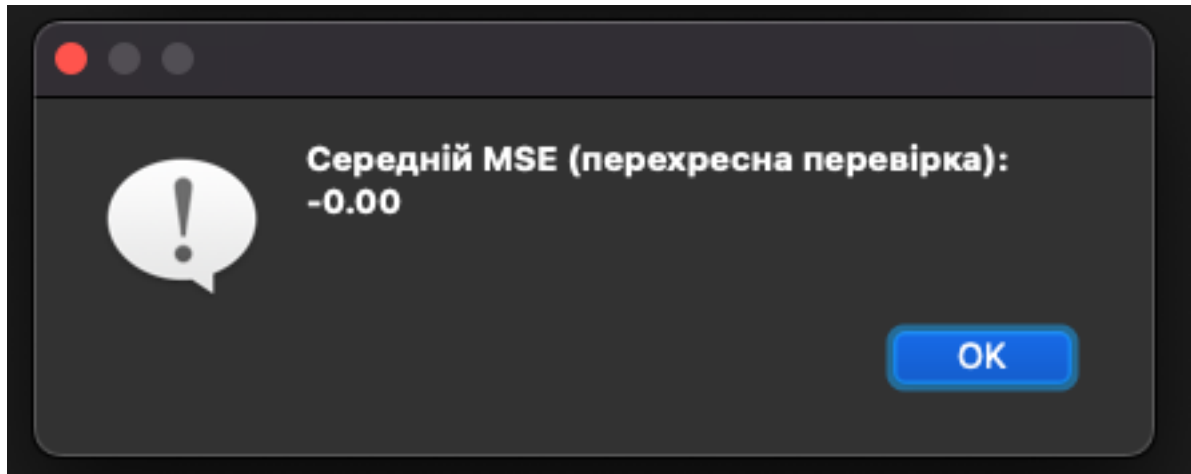


Рисунок 3.10 – Скріншот валідації

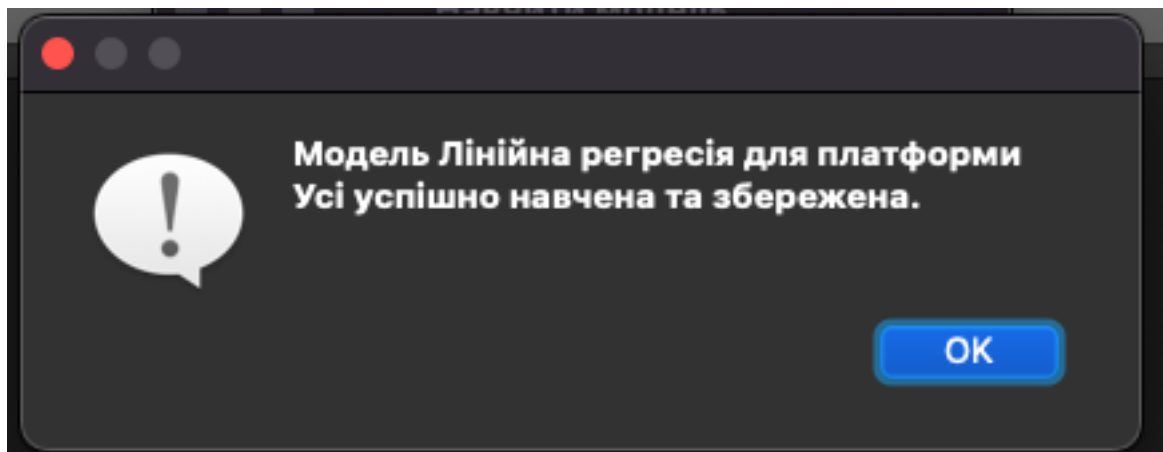
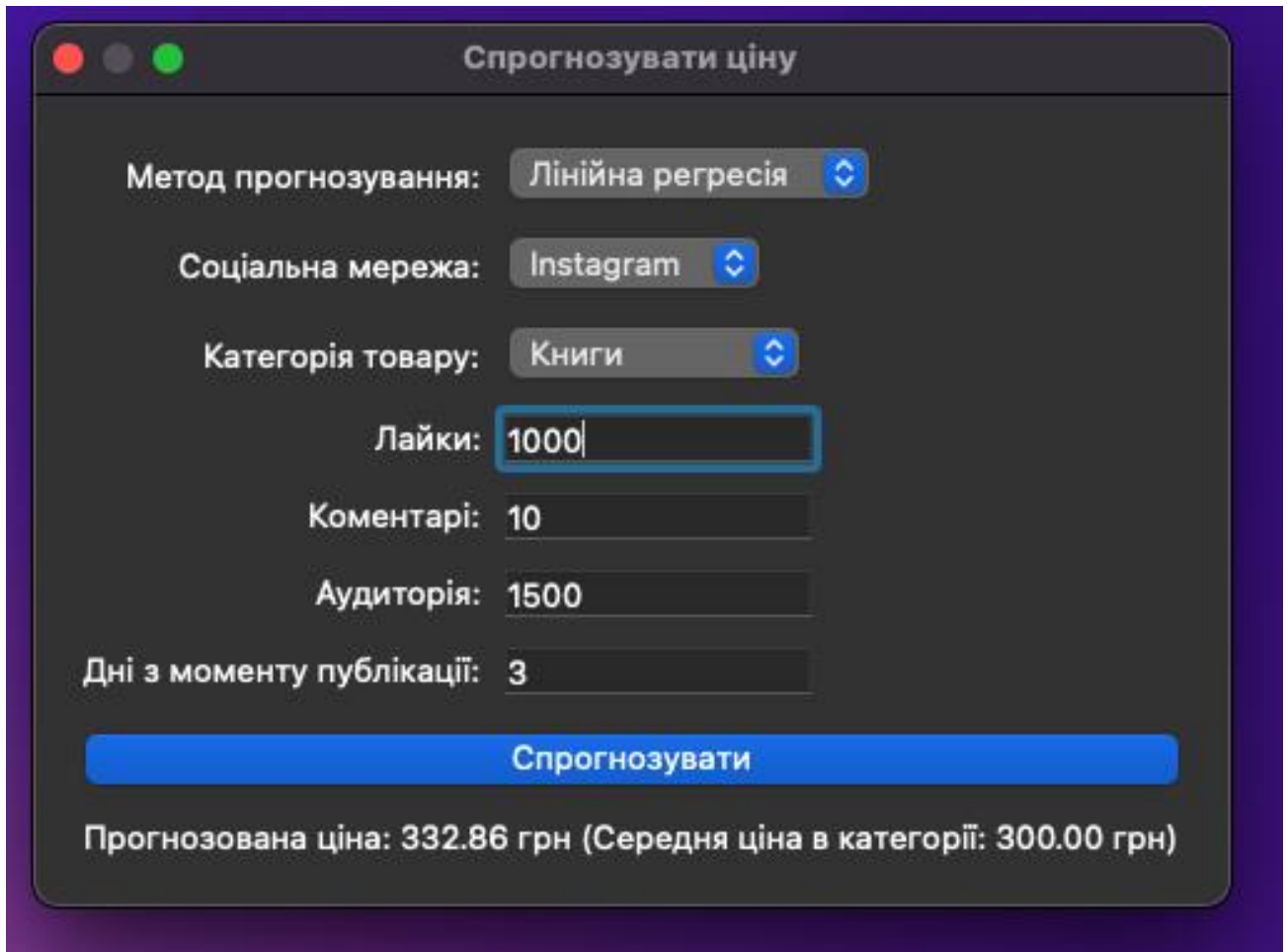


Рисунок 3.11 – Скріншот після успішного навчання моделі

### **Прогнозування цін та оцінка результатів**

Після навчання моделей було здійснено прогнозування цін для обраних товарів. Прогнозовані ціни порівнювалися з фактичними цінами, отриманими з інтернет-магазинів. Це дозволило оцінити точність моделі та виявити, що

алгоритми Random Forest показали найкращі результати, забезпечуючи найменшу середню похибку.



Спрогнозувати ціну

Метод прогнозування: Лінійна регресія

Соціальна мережа: Instagram

Категорія товару: Книги

Лайки: 1000

Коментарі: 10

Аудиторія: 1500

Дні з моменту публікації: 3

Спрогнозувати

Прогнозована ціна: 332.86 грн (Середня ціна в категорії: 300.00 грн)

Рисунок 3.12 - Скріншот інтерфейсу з результатами прогнозування цін

### Аналіз результатів

Аналіз результатів експерименту продемонстрував, що розроблена модель на основі метрик соціальних мереж є ефективною для прогнозування цін на товари. Основними факторами, що впливають на точність прогнозу, є кількість вподобань, коментарів та розмір аудиторії. Ці фактори мають прямий кореляційний зв'язок з ціною товару, що дозволяє використовувати їх як надійні індикатори попиту та цінової політики.

### **3.3 Практичні рекомендації для користувачів системи прогнозування в частині виявлення тенденцій у ціновій динаміці**

Система прогнозування цін на основі метрик соціальних мереж має значний потенціал для покращення стратегій ціноутворення та маркетингу. У цьому підрозділі наведено практичні рекомендації для користувачів щодо виявлення та аналізу тенденцій у ціновій динаміці.

#### **Використання прогнозованих даних для коригування цін**

Результати прогнозування цін слід використовувати для коригування цінової політики компанії. Прогнозовані ціни, що генеруються системою, можуть слугувати основою для встановлення конкурентних цін. Важливо регулярно перевіряти точність прогнозів шляхом порівняння з фактичними цінами, щоб вчасно коригувати моделі прогнозування.

#### **Аналіз тенденцій у динаміці цін**

Для ефективного використання системи слід здійснювати регулярний аналіз тенденцій у динаміці цін на товари. Це дозволить виявити сезонні коливання, вплив маркетингових кампаній та інших факторів на ціни. Використовуючи дані з соціальних мереж, можна визначити, які події або кампанії призводять до збільшення попиту та, відповідно, до змін у ціновій динаміці.

#### **Виявлення ключових факторів впливу на ціни**

Система дозволяє визначити ключові фактори, що впливають на ціни товарів. Аналіз даних допомагає виявити, які показники соціальних мереж найбільше корелюють з цінами. Це можуть бути кількість вподобань, коментарів або розмір аудиторії. Виявлення таких факторів дозволяє компаніям ефективніше

планувати маркетингові стратегії та коригувати цінову політику відповідно до поточних ринкових умов.

### **Постійне вдосконалення моделі**

Для підтримання високої точності прогнозів необхідно постійно вдосконалювати модель прогнозування. Це включає оновлення бази даних, переналаштування параметрів моделі та впровадження нових алгоритмів машинного навчання. Регулярне вдосконалення моделі забезпечить її адаптацію до змін у ринкових умовах та нових викликів.

### **Використання системи для стратегічного планування**

Окрім щоденних операційних завдань, система прогнозування може бути використана для стратегічного планування. Аналіз довгострокових тенденцій у ціновій динаміці дозволяє компаніям приймати обґрунтовані рішення щодо розвитку продуктового асортименту, планування маркетингових кампаній та оптимізації цінової політики.

Ці рекомендації спрямовані на максимальне використання потенціалу системи прогнозування цін на основі метрик соціальних мереж. Впровадження таких підходів дозволить підвищити ефективність бізнесу, забезпечити конкурентоспроможність товарів та адаптуватися до динамічних змін ринку.

### **Висновки за розділом 3**

Розроблена система включає кілька ключових компонентів: збір та зберігання даних, навчання моделей, прогнозування цін та аналіз результатів. Завдяки використанню сучасних технологій, таких як бібліотеки PyQt5, pandas, sklearn та sqlite3, вдалося створити інтуїтивний та функціональний інтерфейс користувача, який забезпечує зручність введення даних, навчання моделей та прогнозування цін.

Експериментальне дослідження підтвердило ефективність розробленої моделі. Алгоритми лінійної регресії, дерева рішень та Random Forest були протестовані на реальних даних, що дало змогу оцінити їх точність та стійкість. Найкращі результати показала модель на основі Random Forest, яка забезпечила найменшу середню похибку прогнозування.

Практичні рекомендації для користувачів системи підкреслюють важливість регулярного аналізу тенденцій у цінній динаміці, моніторингу ключових факторів, що впливають на ціни, та постійного вдосконалення моделей прогнозування. Використання системи для стратегічного планування дозволяє компаніям приймати обґрунтовані рішення щодо розвитку продуктового асортименту та оптимізації цінової політики.

Таким чином, розроблена система прогнозування цін на основі метрик соціальних мереж є потужним інструментом, який може бути інтегрований у системи електронної комерції для підвищення точності прогнозів, адаптації до ринкових умов та забезпечення конкурентоспроможності товарів.

## ВИСНОВКИ

Результатом цієї роботи є модель прогнозування ціни на товар на основі метрик соціальних мереж.

Використання даних з соціальних мереж для прогнозування цін на товари в сучасній економіці відіграє важливу роль. Завдяки збору та аналізу великого обсягу інформації, що генерується користувачами, компанії можуть отримати цінні інсайти про споживчі уподобання, тренди та зміни в попиті. Це відкриває можливості для швидкої адаптації стратегій ціноутворення та реагування на ринкові зміни у реальному часі. Однак такий підхід супроводжується викликами та обмеженнями. Наприклад, дані з соціальних мереж можуть бути неоднорідними та не завжди повністю репрезентативними. Можливість маніпулювати даними та недостатня об'єктивність відгуків користувачів може призвести до неточностей у прогнозуванні. Крім того, аналіз даних з соціальних мереж вимагає значної обробки та обробки, оскільки ці дані часто представлені у неструктурованому форматі. Це може вимагати значних витрат часу та ресурсів для виявлення корисної інформації та побудови ефективних моделей прогнозування.

Процес тестування та впровадження моделі прогнозування цін включає кілька етапів для забезпечення її ефективності та точності. Після розробки моделі важливо провести тестування на історичних даних для перевірки її прогностичної здатності. Після успішного тестування модель може бути впроваджена в робоче середовище, інтегруючись з існуючими системами електронної комерції. Потім важливо відслідковувати продуктивність та ефективність моделі в реальному часі, регулярно моніторивши її прогнози та порівнюючи їх з фактичними цінами. З урахуванням динаміки ринку модель потребує постійного оновлення та вдосконалення, щоб забезпечити її високу ефективність у довгостроковій перспективі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білоус К. С. Сутність маркетингу соціальних мереж : thesis. 2019. [Електронний ресурс]. – режим доступу: URL: <https://er.knutd.edu.ua/handle/123456789/12754> (дата звернення – 22.03.2024).
2. В.О Касьяненко, Л.В.Старченко. Моделювання та прогнозування економічних процесів. Конспект лекцій : навч. посіб. Київ : Унів. кн., 2023. 185 с.
3. Експлораторний та конфірмаційний підходи. [Електронний ресурс]. – режим доступу: URL: <https://subjectum.eu/sociology/dict/132.html> (дата звернення – 02.04.2024).
4. Медіамейкер. [Електронний ресурс]. – режим доступу: URL: <https://mediamaker.me/news/buty-soboyu-j-ne-pereproshuvaty-za-cze-yaki-trendy-dominuvatymut-v-instagram-2024-roku/> (дата звернення – 19.03.2024).
5. Непрямі методи дослідження цін. [Електронний ресурс]. – режим доступу: URL: [https://pidru4niki.com/12000725/marketing/nepryami\\_metodi\\_doslidzhennya\\_tsin](https://pidru4niki.com/12000725/marketing/nepryami_metodi_doslidzhennya_tsin) (дата звернення – 19.03.2024).
6. Олексій Васильєв. Програмування мовою Python. Навчальна книга – Богдан, 2019. 540 с.
7. Смолянчук О. В. Соціальні мережі: значення для маркетингової діяльності. Економіка та держава. 2013. № 3. С. 91–93.
8. Що таке соціальні мережі? [Електронний ресурс]. – режим доступу: URL: <https://futurenow.com.ua/shho-take-sotsialni-merezhi-vydy-klasyfikatsiya-bezpeka/> (дата звернення – 19.03.2024).
9. Ярмолюк О. Я., Фісун Ю. В., Шаповалова А. А. СОЦІАЛЬНІ МЕРЕЖІ ЯК СУЧАСНИЙ ІНСТРУМЕНТ ПРОСУВАННЯ. Підприємництво та інновації. 2020. № 11-2. С. 62–65. [Електронний ресурс]. – режим доступу: URL: <https://doi.org/10.37320/2415-3583/11.28> (дата звернення – 22.03.2024).

10. Asur S., Huberman B. A. Predicting the future with social media // 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. 2010. Vol. 1. P. 492–499. IEEE.
11. Bollen J., Huina M. Twitter mood as a stock market predictor // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2011. 41(6). P. 1160–1170.
12. Bollen J., Mao H., Zeng X. Twitter mood predicts the stock market // Journal of Computational Science. 2011. 2(1). P. 1–8.
13. Gandomi A., Haider M. Beyond the hype: Big data concepts, methods, and analytics // International Journal of Information Management. 2015. 35(2). P. 137–144.
14. Kearney M. W. rtweet: Collecting and analyzing Twitter data // Journal of Open Source Software. 2018. 3(22). P. 182.
15. McKinney W. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc., 2017.
16. Pang B., Lee L. Opinion mining and sentiment analysis // Foundations and Trends in Information Retrieval. 2008. 2(1-2). P. 1–135.
17. Raschka S., Mirjalili V. Python machine learning. Packt Publishing Ltd., 2017.
18. Saad E., Yang J. Using machine learning algorithms for predicting stock market prices based on social media sentiment analysis // 2019 5th International Conference on Advances in Computing, Communication & Automation (ICACCA). 2019. P. 1–6. IEEE.
19. Stieglitz S., Mirbabaie M., Ross B., Neuberger C. Social media analytics – Challenges in topic discovery, data collection, and data preparation // International Journal of Information Management. 2018. 39. P. 156–168.
20. Tan C. H., Lee C., Tan C. Analyzing the effects of social media on stock price movements // Expert Systems with Applications. 2015. 42(1). P. 408–414.
21. VanderPlas J. Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc., 2016. 548 p.
22. Alchin J. Pro Python 3: Features and Tools for Professional Development. Apress, 2018. 362 p.

23. Barry P. Head *First Python: A Brain-Friendly Guide*. O'Reilly Media, Inc., 2016. 624 p.
24. Beazley D. M., Jones B. K. *Python Cookbook: Recipes for Mastering Python 3*. O'Reilly Media, Inc., 2013. 706 p.
25. Lutz M. *Learning Python*. O'Reilly Media, Inc., 2013. 1648 p.
26. Sweigart A. *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press, 2015. 504 p.

## ДОДАТКИ

### Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**  
Галузь знань: 12 – Інформаційні технології  
Спеціальність: 123 – Комп'ютерна інженерія.

ЗАТВЕРДЖУЮ  
Завідувач кафедри теоретичної  
та прикладної системотехніки  
д.т.н., проф. Шматков С. І.  
«21» грудня 2024 року

### З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ

**БІЛОУСА АНДРІЯ МИКОЛАЙОВИЧА**

(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Модель прогнозування цін на товари на основі метрик соціальних мереж»**

керівник роботи Чуб Ольга Ігорівна, к.е.н, доцент,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити

- 1) Постановка задачі прогнозування цін на товари з використанням метрик соціальних мереж.
- 2) Аналіз існуючих методів прогнозування цін та впливу соціальних мереж.
- 3) Вибір та обґрунтування алгоритмів для аналізу даних з соціальних мереж.
- 4) Розробка математичної моделі для прогнозування цін на товари.
- 5) Розробка програмно-алгоритмічної моделі для аналізу метрик соціальних мереж.
- 6) Тестування моделі та аналіз отриманих результатів.

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Підбір наукової літератури та джерел про соціальні мережі	21.12.2023 - 25.01.2024
2	Огляд сучасних методів прогнозування цін	19.12.2023 - 2.01.2024
3	Аналіз інструментальних засобів для обробки даних з соціальних мереж	2.01.2024 - 2.02.2024
4	Розробка комп'ютерної моделі для прогнозування цін на товари	2.01.2024 - 2.02.2024
5	Тестування та апробація комп'ютерної моделі.	3.02.2024 - 30.03.2024
6	Корегування моделі після тестування та оформлення пояснювальної записки	3.03.2024 - 30.04.2024
7	Розробка пояснювальної записки.	31.03.2024 - 27.05.2024

## 5. Дата видачі завдання 21.12.2023

Студент

А. М. Білоус

ініціали, прізвище



підпис

Керівник роботи

О. І. Чуб

ініціали, прізвище



підпис

Додаток Б

Затверджую

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**Технічне завдання  
на розробку прототипу «МАШИНА МОДЕЛЬ ПРОГНОЗУВАННЯ ЦІН НА  
ТОВАРИ ЗА ДОПОМОГОЮ МЕТРИК СОЦІАЛЬНИХ МЕРЕЖ».**

1.	Введення	1.1. Назва роботи – «Система прогнозування цін на товари за допомогою метрик соціальних мереж». 1.2. Галузь застосування: 12 – Інформаційні технології.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп’ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/909 від «03» травня 2024 року (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: Створення програмного рішення для аналізу та прогнозування цін на товари на основі аналізу даних з соціальних мереж. 3.2. Призначення розробки: Забезпечення інструментарію для компаній, що спеціалізуються на електронній комерції, для покращення стратегій ціноутворення і маркетингу. 3.3. Вхідні дані: Дані про користувачів і їхні вподобання з соціальних мереж, історичні дані про ціни, реакції на маркетингові кампанії. 3.4. Вихідні дані розробки: Моделі прогнозування цін, звіти про ефективність прогнозів, рекомендації для коригування маркетингових стратегій.

4.	Технічні вимоги до програмного виробу	<p>4.1. Функціональні вимоги:</p> <ul style="list-style-type: none"> <li>• Автоматичний збір даних з соціальних мереж.</li> <li>• Аналіз впливу метрик соціальних мереж на ціни товарів.</li> <li>• Розробка алгоритмів машинного навчання для прогнозування цін.</li> <li>• Візуалізація даних та результатів аналізу.</li> </ul> <p>4.2. Нефункціональні вимоги:</p> <ul style="list-style-type: none"> <li>• Висока надійність і точність прогнозів.</li> <li>• Швидкодія і масштабованість системи.</li> <li>• Інтуїтивно зрозумілий інтерфейс для користувачів.</li> </ul> <p>4.3. Вимоги до інтеграції:</p> <ul style="list-style-type: none"> <li>• Інтеграція з існуючими CRM та ERP системами.</li> <li>• Сумісність з популярними платформами електронної комерції.</li> </ul> <p>4.4. Вимоги до безпеки:</p> <ul style="list-style-type: none"> <li>• Захист персональних даних згідно з GDPR та іншими регуляторними вимогами.</li> </ul> <p>Запобігання несанкціонованому доступу до аналітичних результатів.</p>	
5.	Вимоги до програмної документації	<p>Документацією до виробу «Система прогнозування цін на товари за допомогою метрик соціальних мереж» вважати:</p> <p>5.1. Опис основних вимог та функціональності системи (розділ 4 пояснювальної записки).</p> <p>5.2. Програма і методика випробувань розробленої програми (додаток В).</p> <p>5.3. Опис реалізованого програмного продукту (розділ 3 пояснювальної записки).</p> <p>5.4. Джерела базової інформації.</p>	
6.	Стадії і етапи розробки	Дата	Назва етапу
		14.02.2024 - 28.02.2024	Огляд існуючих моделей прогнозування цін на товари та вивчення впливу метрик соціальних мереж.
		29.02.2024 - 15.03.2024	Аналіз методів збору даних з соціальних мереж та їх обробки для використання у прогнозуванні цін.

		1.03.2024 - 15.03.2024	Вибір технологій та інструментів для розробки моделі прогнозування цін.
		15.03.2024 - 15.04.2024	Проектування архітектури системи та розробка базового прототипу.
		25.03.2024 - 30.04.2024	Інтеграція зібраних даних з соціальних мереж і реалізація алгоритмів машинного навчання для прогнозування.
		15.05.2024 - 31.05.2024	Тестування та оцінка прототипу, порівняння результатів з очікуваними показниками та аналіз отриманих висновків.
		31.05.2024	Оформлення звіту за результатами переддипломної практики та фінальна підготовка презентації.
		31.05.2024	Представлення кваліфікаційної роботи керівнику та рецензенту.
8.	Порядок контролю і приймання програмного продукту (моделі)		<ol style="list-style-type: none"> <li>1. Перевірку ходу розробки прототипу моделі виконувати раз в 3 тижні.</li> <li>2. Захист розробленої моделі провести на засіданні Атестаційної комісії.</li> <li>3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді.</li> </ol>

Виконавець

Студент групи КІ-41

Білоус А. М.



Замовник

канд. екон. наук

Чуб О. І.



## Додаток В

**Програма і методика випробувань програмного виробу****«МОДЕЛЬ ПРОГНОЗУВАННЯ ЦІН НА ТОВАРИ НА ОСНОВІ МЕТРИК СОЦІАЛЬНИХ МЕРЕЖ»****1. Об'єкт випробувань**

1. **Назва:** модель прогнозування цін на товари на основі метрик соціальних мереж

2. **Галузь застосування:** ІТ – Інформаційні технології.

3. **Основні характеристики об'єкта випробувань:**

- Використання метрик соціальних мереж (лайки, коментарі, охоплення тощо).

Цю модель можна представити у вигляді формули:

$$\text{Price} = \beta_0 + \beta_1 * \text{Days} + \beta_2 * \text{Likes} + \beta_3 * \text{Comments} + \beta_4 * \text{Audience}.$$

Ця формула представляє модель лінійної регресії, де кожний коефіцієнт  $\beta$  відображає вплив відповідної змінної на ціну товару. Розглянемо кожен елемент формули:

**Price:** це залежна змінна, яку ми намагаємося передбачити.

**$\beta_0$ :** це незалежний член, або "перетин", що відображає базовий рівень ціни, коли всі інші змінні дорівнюють нулю.

**$\beta_1, \beta_2, \beta_3, \beta_4$ :** це коефіцієнти регресії, які визначають вплив кожної незалежної змінної (Days, Likes, Comments, Audience) на ціну товару.

**Days:** це кількість днів, пройдених з моменту публікації посту. Ця змінна відображає часовий ефект на ціну.

**Likes:** це кількість лайків, які отримав пост в Instagram. Він відображає рівень зацікавленості аудиторії в продукті.

Comments: це кількість коментарів під постом. Він може вказувати на активність аудиторії та їхні враження від продукту.

Audience: це розмір аудиторії, яка підписана на аккаунт. Велика аудиторія може вказувати на широке охоплення та більшу популярність продукту.

Отже, дана формула дозволяє оцінити вплив різних факторів, таких як кількість днів з моменту публікації посту, кількість лайків та коментарів, а також розмір аудиторії на ціну товару, що допомагає зрозуміти, як ці метрики впливають на цінову стратегію.

- Генерація прогнозів цін на основі аналізу даних.

Відповідно до цих даних, було обрано значення коефіцієнтів:  $\beta_1 = -0.05$  (з кожним днем ціна знижується на 5 копійок),  $\beta_2 = 0.001$  (кожен лайк піднімає ціну товару на 0.1 копійку),  $\beta_3 = 0.1$  (кожен коментар піднімає ціну на 10 копійок),  $\beta_4 = 0.00005$  (кожні 10 тис. підписників піднімають ціну товару на 5 гривень).

## 2. Мета випробувань

Перевірити відповідність функціональних можливостей системи заявленим вимогам у технічному завданні. Оцінити точність прогнозування цін та стабільність роботи системи при обробці різних наборів даних.

## 3. Загальні положення

### 3.1 Підстави для проведення випробувань:

Наказ про призначення атестаційної комісії.

### 3.2. Місце та тривалість випробувань:

Випробування проводяться на базі комп'ютерного класу кафедри під час роботи атестаційної комісії.

### 3.3. Обсяг випробувань:

Обсяг випробувань відповідає програмі та методиці випробувань, яка включає тестування системи на точність прогнозів, функціональну відповідність, стійкість та відновлення після помилок.

### **3.4. Організації, які беруть участь у випробуваннях:**

Атестаційна комісія, розробник, а також представники замовника або зацікавлені сторони.

## **4. Вимоги до програми або програмного виробу**

### **4.1. Функціональні вимоги:**

- Збір та аналіз даних із соціальних мереж.
- Використання метрик соціальних мереж для моделювання цінових трендів.
- Генерація прогнозів цін на товари.

### **4.2. Нефункціональні вимоги:**

- Висока точність прогнозів.
- Надійність та стабільність системи під час великих обсягів обробки даних.
- Простий інтерфейс для кінцевих користувачів.

### **4.3. Вимоги до інтеграції:**

- Сумісність із різними платформами електронної комерції.
- Інтеграція з іншими аналітичними інструментами.

### **4.4. Вимоги до безпеки:**

- Захист даних користувачів.
- Запобігання несанкціонованому доступу.

## **5. Вимоги до програмної документації**

5.1. Технічне завдання на розробку системи.

5.2. Опис реалізованого прототипу системи.

5.3. Джерела базової інформації.

## **6. Засоби та порядок випробувань**

### **6.1. Засоби випробувань:**

Ноутбуки та ПК із встановленим аналітичним програмним забезпеченням та візуалізаційними інструментами.

## **6.2. Порядок проведення випробувань:**

### **Перший етап:**

Перевірка комплектності та якості програмної документації відповідно до ГОСТ 34.602-89.

### **Другий етап:**

- Запуск системи та виконання тестів на різних наборах даних.
- Вимірювання точності моделі прогнозування цін.
- Перевірка відновлення системи після помилок.

## **7. Проведення випробувань**

### **7.1. Збір та аналіз даних:**

Під час випробувань система має вміти коректно завантажувати дані з соціальних мереж та аналізувати їх відповідно до алгоритму прогнозування.

Приклад зчитування даних з таблиці

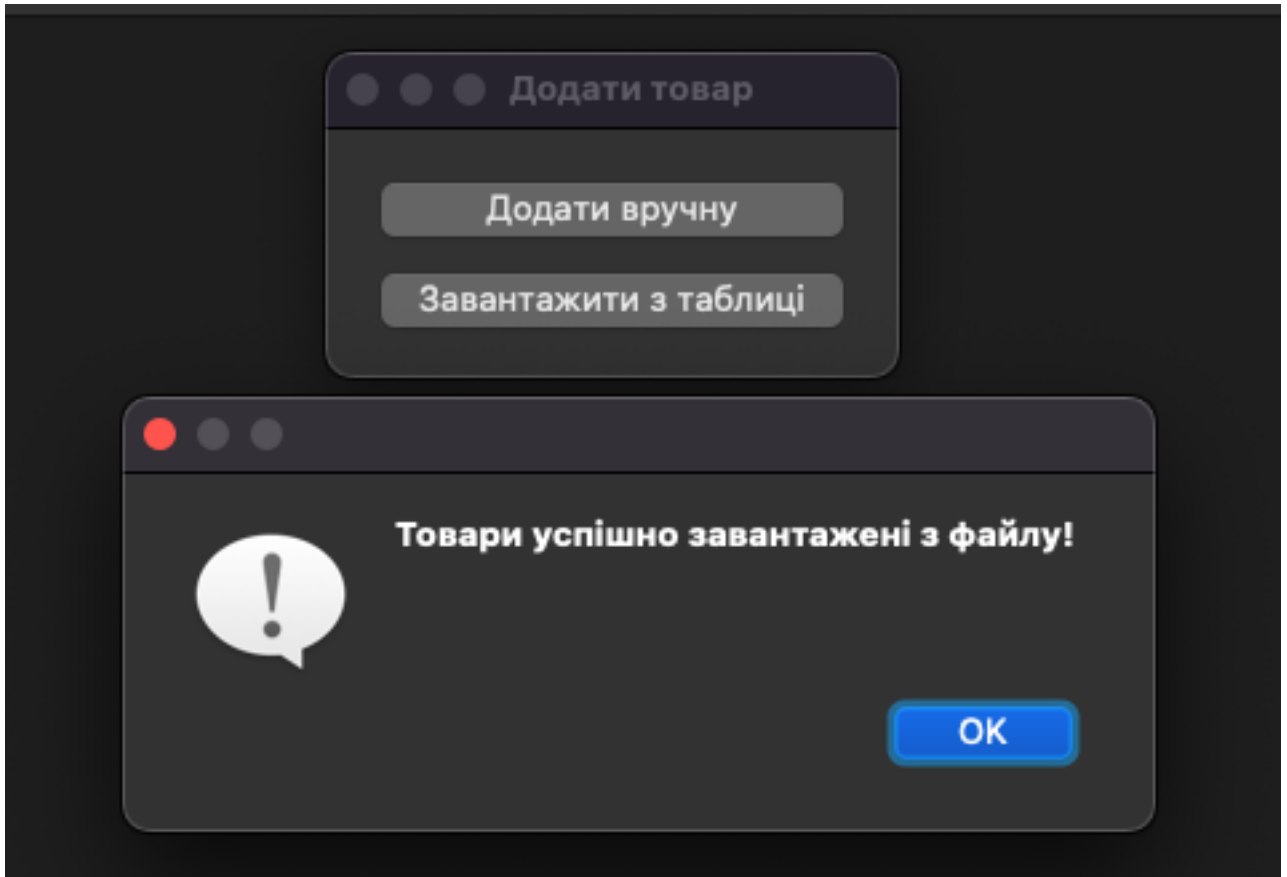
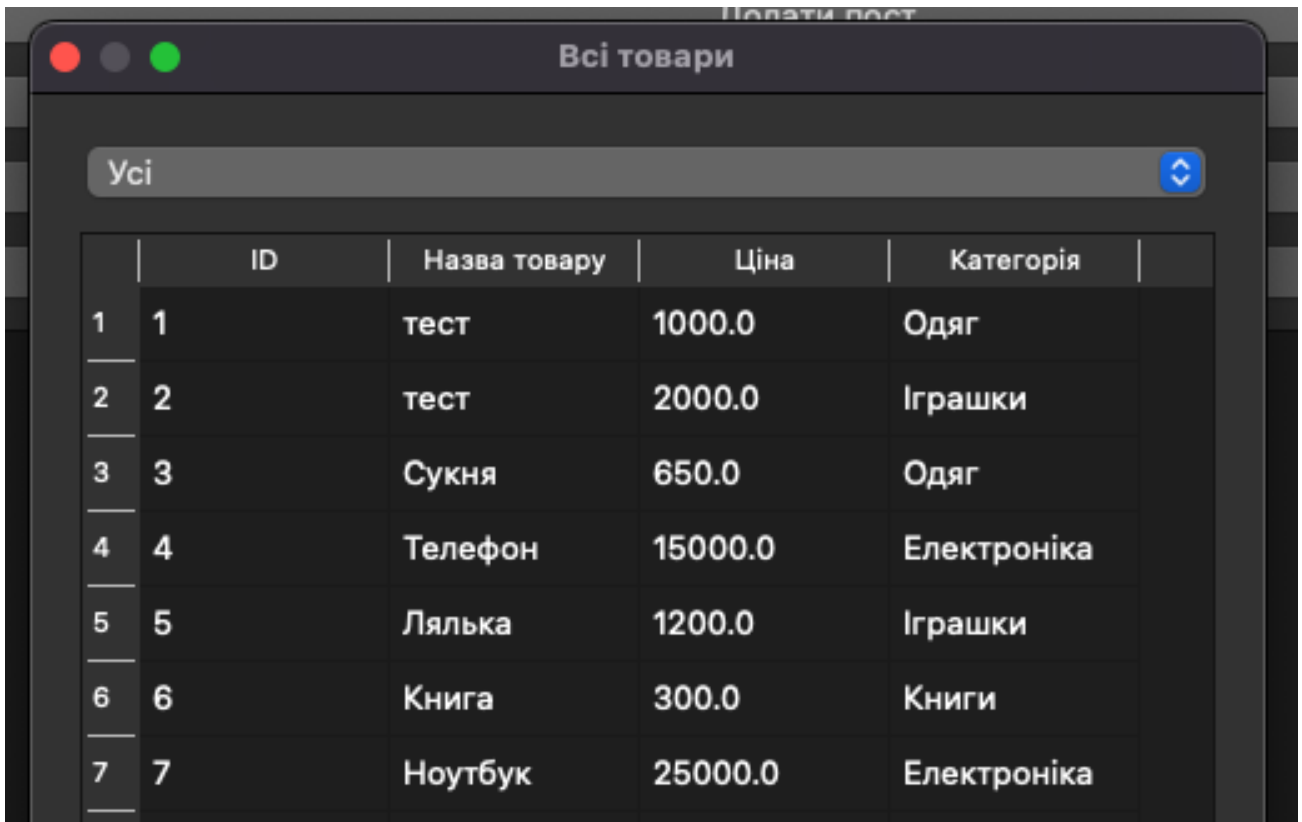


Рисунок В.1 – Скріншот успішного завантаження товарів



	ID	Назва товару	Ціна	Категорія
1	1	тест	1000.0	Одяг
2	2	тест	2000.0	Іграшки
3	3	Сукня	650.0	Одяг
4	4	Телефон	15000.0	Електроніка
5	5	Лялька	1200.0	Іграшки
6	6	Книга	300.0	Книги
7	7	Ноутбук	25000.0	Електроніка

Рисунок В.2 – Скріншот таблиці товарів

## 7.2. Навчання моделі:

Система повинна мати інтерфейс для навчання моделі різними методами та для різних соціальної мережі

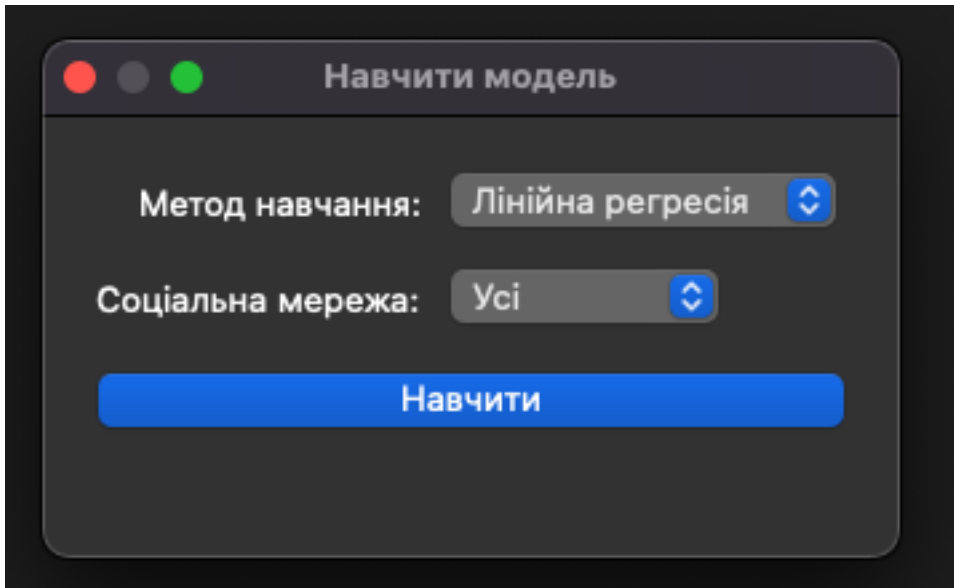


Рисунок В.3 – Інтерфейс для додавання навчання моделі

### 7.3. Результати випробувань:

Система повинна на прогноровані ціни для набору товарів, а також виводити результати у зрозумілій та зручній формі. Результати прогнозування перевіряються на відповідність заявленим критеріям точності та функціональності.

Приклад генерації цін на основі даних з таблиці

The image shows a software window titled "Спрогнозувати ціну" (Predict Price) with a dark grey background and a blue border. The window contains several input fields and a button. The inputs are: "Метод прогнозування" (Prediction Method) set to "Лінійна регресія" (Linear Regression); "Соціальна мережа" (Social Network) set to "Instagram"; "Категорія товару" (Product Category) set to "Книги" (Books); "Лайки" (Likes) set to "1000"; "Коментарі" (Comments) set to "10"; "Аудиторія" (Audience) set to "1500"; and "Дні з моменту публікації" (Days since publication) set to "3". A blue button labeled "Спрогнозувати" (Predict) is positioned below the inputs. At the bottom of the window, the predicted price is displayed as "Прогнозована ціна: 332.86 грн (Середня ціна в категорії: 300.00 грн)".

Field	Value
Метод прогнозування	Лінійна регресія
Соціальна мережа	Instagram
Категорія товару	Книги
Лайки	1000
Коментарі	10
Аудиторія	1500
Дні з моменту публікації	3

Прогнозована ціна: 332.86 грн (Середня ціна в категорії: 300.00 грн)

Рисунок В.4 – результат прогнозування ціни

**ЛІСТИНГ ПРОГРАМНОГО КОДУ**

```
import sys
import os
from PyQt5.QtWidgets import QApplication, QMainWindow, QLabel,
QVBoxLayout, QPushButton, QFileDialog, QTableWidgetItem, \
    QTableWidgetItem, QWidget, QDialog, QFormLayout, QLineEdit, QComboBox,
QMessageBox
from PyQt5.QtCore import Qt
import pandas as pd
import numpy as np
import sqlite3
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score
import pickle

class PricePredictionApp(QMainWindow):
    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.setWindowTitle("Система прогнозування цін на товари")
        self.setGeometry(100, 100, 800, 600)
```

```
container = QWidget()
self.setCentralWidget(container)
layout = QVBoxLayout(container)

self.addProductButton = QPushButton('Додати товар', self)
self.addProductButton.clicked.connect(self.add_product_window)
layout.addWidget(self.addProductButton)

self.viewProductsButton = QPushButton('Переглянути товари', self)
self.viewProductsButton.clicked.connect(self.view_products_window)
layout.addWidget(self.viewProductsButton)

self.addPostButton = QPushButton('Додати пост', self)
self.addPostButton.clicked.connect(self.add_post_window)
layout.addWidget(self.addPostButton)

self.viewPostsButton = QPushButton('Переглянути всі пости', self)
self.viewPostsButton.clicked.connect(self.view_posts_window)
layout.addWidget(self.viewPostsButton)

self.trainModelButton = QPushButton('Навчити модель', self)
self.trainModelButton.clicked.connect(self.train_model_window)
layout.addWidget(self.trainModelButton)

self.predictPricesButton = QPushButton('Спрогнозувати ціни', self)
self.predictPricesButton.clicked.connect(self.predict_price_window)
layout.addWidget(self.predictPricesButton)
```

```

self.tableWidget = QTableWidgetItem()
layout.addWidget(self.tableWidget)

self.resultLabel = QLabel('Результати з\'являться тут', self)
layout.addWidget(self.resultLabel)

self.db_init()

def db_init(self):
    self.conn = sqlite3.connect('price_prediction.db')
    self.cursor = self.conn.cursor()
    self.cursor.execute("""CREATE TABLE IF NOT EXISTS products
                          (id INTEGER PRIMARY KEY, name TEXT, price REAL,
category TEXT)""")
    self.cursor.execute("""CREATE TABLE IF NOT EXISTS posts
                          (id INTEGER PRIMARY KEY, platform TEXT, likes INTEGER,
comments INTEGER,
                          audience INTEGER, days_since_publication INTEGER,
product_id INTEGER,
                          FOREIGN KEY(product_id) REFERENCES products(id))""")
    self.conn.commit()

def add_product_window(self):
    dialog = QDialog(self)
    dialog.setWindowTitle("Додати товар")
    layout = QVBoxLayout(dialog)

    manual_button = QPushButton("Додати вручну", dialog)

```

```
manual_button.clicked.connect(lambda: self.manual_add_product(dialog))
layout.addWidget(manual_button)
```

```
load_button = QPushButton("Завантажити з таблиці", dialog)
load_button.clicked.connect(lambda: self.load_products_from_file(dialog))
layout.addWidget(load_button)
```

```
dialog.exec_()
```

```
def manual_add_product(self, parent_dialog):
    dialog = QDialog(parent_dialog)
    dialog.setWindowTitle("Додати товар вручну")
    layout = QFormLayout(dialog)

    name_edit = QLineEdit(dialog)
    price_edit = QLineEdit(dialog)
    category_edit = QComboBox(dialog)
    category_edit.addItems(["Одяг", "Електроніка", "Іграшки", "Книги", "Інше"])

    layout.addRow("Назва товару:", name_edit)
    layout.addRow("Поточна ціна:", price_edit)
    layout.addRow("Категорія:", category_edit)

    add_button = QPushButton("Додати", dialog)
    add_button.clicked.connect(lambda: self.save_product(name_edit.text(),
price_edit.text(), category_edit.currentText(), dialog))
    layout.addRow(add_button)
```

```

dialog.exec_()

def save_product(self, name, price, category, dialog):
    try:
        self.cursor.execute("INSERT INTO products (name, price, category)
VALUES (?, ?, ?)", (name, float(price), category))
        self.conn.commit()
        QMessageBox.information(self, "Інформація", "Товар успішно доданий!")
        dialog.accept()
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка при додаванні товару:
{e}")

def load_products_from_file(self, parent_dialog):
    options = QFileDialog.Options()
    file_path, _ = QFileDialog.getOpenFileName(self, "Відкрити Excel файл", "",
"Excel Files (*.xlsx);;All Files (*)", options=options)
    if file_path:
        data = read_products_from_excel(file_path)
        if data is not None:
            try:
                for _, row in data.iterrows():
                    self.cursor.execute("INSERT INTO products (name, price, category)
VALUES (?, ?, ?)",
                                        (row['Назва товару'], row['Ціна'], row['Категорія']))
                    self.conn.commit()
                    QMessageBox.information(self, "Інформація", "Товари успішно
завантажені з файлу!")

```

```

        parent_dialog.accept()
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка при завантаженні
товарів: {e}")

```

```

def view_products_window(self):
    dialog = QDialog(self)
    dialog.setWindowTitle("Всі товари")
    layout = QVBoxLayout(dialog)

    category_combo = QComboBox(dialog)
    category_combo.addItem("Усі") + [row[0] for row in
self.cursor.execute("SELECT DISTINCT category FROM products").fetchall()]
    category_combo.currentTextChanged.connect(lambda:
self.display_products_by_category(category_combo.currentText(), dialog))
    layout.addWidget(category_combo)

    self.productTable = QTableWidgetItem(dialog)
    layout.addWidget(self.productTable)

    self.update_product_button = QPushButton("Оновити товар", dialog)
    self.update_product_button.clicked.connect(self.update_product)
    layout.addWidget(self.update_product_button)

    self.delete_product_button = QPushButton("Видалити товар", dialog)
    self.delete_product_button.clicked.connect(self.delete_product)
    layout.addWidget(self.delete_product_button)

```

```

self.display_products_by_category("Yci", dialog)

dialog.exec_()

def display_products_by_category(self, category, dialog):
    if category == "Yci":
        query = "SELECT id, name, price, category FROM products"
        rows = self.cursor.execute(query).fetchall()
    else:
        query = "SELECT id, name, price, category FROM products WHERE
category=?"
        rows = self.cursor.execute(query, (category,)).fetchall()

    self.productTable.setRowCount(len(rows))
    self.productTable.setColumnCount(4)
    self.productTable.setHorizontalHeaderLabels(["ID", "Назва товару", "Ціна",
"Категорія"])

    for i, row in enumerate(rows):
        for j, val in enumerate(row):
            self.productTable.setItem(i, j, QTableWidgetItem(str(val)))

def update_product(self):
    selected_items = self.productTable.selectedItems()
    if selected_items:
        product_id = int(selected_items[0].text())
        current_name = selected_items[1].text()
        current_price = selected_items[2].text()

```

```

current_category = selected_items[3].text()

dialog = QDialog(self)
dialog.setWindowTitle("Оновити товар")
layout = QFormLayout(dialog)

name_edit = QLineEdit(dialog)
name_edit.setText(current_name)
price_edit = QLineEdit(dialog)
price_edit.setText(current_price)
category_edit = QComboBox(dialog)
category_edit.addItems(["Одяг", "Електроніка", "Іграшки", "Книги",
"Інше"])
category_edit.setCurrentText(current_category)

layout.addRow("Назва товару:", name_edit)
layout.addRow("Поточна ціна:", price_edit)
layout.addRow("Категорія:", category_edit)

update_button = QPushButton("Оновити", dialog)
update_button.clicked.connect(lambda:
self.save_updated_product(product_id, name_edit.text(), price_edit.text(),
category_edit.currentText(), dialog))
layout.addRow(update_button)

dialog.exec_()
else:
    QMessageBox.warning(self, "Помилка", "Будь ласка, оберіть товар для

```

оновлення.")

```
def save_updated_product(self, product_id, name, price, category, dialog):
    try:
        self.cursor.execute("UPDATE products SET name=?, price=?, category=?
WHERE id=?", (name, float(price), category, product_id))
        self.conn.commit()
        QMessageBox.information(self, "Інформація", "Товар успішно
оновлений!")
        dialog.accept()
        self.view_products_window()
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка при оновленні товару:
{e}")
```

```
def delete_product(self):
    selected_items = self.productTable.selectedItems()
    if selected_items:
        product_id = int(selected_items[0].text())
        confirm = QMessageBox.question(self, "Підтвердження", "Ви впевнені, що
хочете видалити цей товар?", QMessageBox.Yes | QMessageBox.No)
        if confirm == QMessageBox.Yes:
            try:
                self.cursor.execute("DELETE FROM products WHERE id=?",
(product_id,))
                self.conn.commit()
                QMessageBox.information(self, "Інформація", "Товар успішно
видалений!")
```

```

        self.view_products_window()
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка при видаленні
товару: {e}")
    else:
        QMessageBox.warning(self, "Помилка", "Будь ласка, оберіть товар для
видалення.")

```

```

def add_post_window(self):

```

```

    dialog = QDialog(self)
    dialog.setWindowTitle("Додати пост")
    layout = QVBoxLayout(dialog)

```

```

    manual_button = QPushButton("Додати вручну", dialog)
    manual_button.clicked.connect(lambda: self.manual_add_post(dialog))
    layout.addWidget(manual_button)

```

```

    load_button = QPushButton("Завантажити з таблиці", dialog)
    load_button.clicked.connect(lambda: self.load_posts_from_file(dialog))
    layout.addWidget(load_button)

```

```

    dialog.exec_()

```

```

def manual_add_post(self, parent_dialog):

```

```

    dialog = QDialog(parent_dialog)
    dialog.setWindowTitle("Додати пост вручну")
    layout = QFormLayout(dialog)

```

```

platform_edit = QComboBox(dialog)
platform_edit.addItems(["Facebook", "Instagram", "Twitter", "Інше"])
likes_edit = QLineEdit(dialog)
comments_edit = QLineEdit(dialog)
audience_edit = QLineEdit(dialog)
days_edit = QLineEdit(dialog)

product_combo = QComboBox(dialog)
product_combo.addItems([row[0] for row in self.cursor.execute("SELECT name
FROM products").fetchall()])

layout.addRow("Платформа:", platform_edit)
layout.addRow("Лайки:", likes_edit)
layout.addRow("Коментарі:", comments_edit)
layout.addRow("Аудиторія:", audience_edit)
layout.addRow("Дні з моменту публікації:", days_edit)
layout.addRow("Товар:", product_combo)

add_button = QPushButton("Додати", dialog)
add_button.clicked.connect(lambda: self.save_post(platform_edit.currentText(),
likes_edit.text(), comments_edit.text(),
audience_edit.text(), days_edit.text(),
product_combo.currentText(), dialog))
layout.addRow(add_button)

dialog.exec_()

```

```

def save_post(self, platform, likes, comments, audience, days, product_name,
dialog):
    try:
        product_id = self.cursor.execute("SELECT id FROM products WHERE
name=?", (product_name,)).fetchone()[0]
        self.cursor.execute("INSERT INTO posts (platform, likes, comments,
audience, days_since_publication, product_id) VALUES (?, ?, ?, ?, ?, ?)",
        (platform, int(likes), int(comments), int(audience), int(days),
product_id))
        self.conn.commit()
        QMessageBox.information(self, "Інформація", "Пост успішно доданий!")
        dialog.accept()
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка при додаванні посту:
{e}")

def load_posts_from_file(self, parent_dialog):
    options = QFileDialog.Options()
    file_path, _ = QFileDialog.getOpenFileName(self, "Відкрити Excel файл", "",
"Excel Files (*.xlsx);;All Files (*)", options=options)
    if file_path:
        data = read_posts_from_excel(file_path)
        if data is not None:
            try:
                for _, row in data.iterrows():
                    product_id = self.cursor.execute("SELECT id FROM products
WHERE name=?", (row['Товар'],)).fetchone()[0]
                    self.cursor.execute("INSERT INTO posts (platform, likes, comments,

```

```

audience, days_since_publication, product_id) VALUES (?, ?, ?, ?, ?, ?)",
                (row['Платформа'], int(row['Лайки']),
int(row['Коментарі']), int(row['Аудиторія']),
                int(row['Дні з моменту публікації']), product_id))
        self.conn.commit()
        QMessageBox.information(self, "Інформація", "Пости успішно
завантажені з файлу!")
        parent_dialog.accept()
    except Exception as e:
        QMessageBox.warning(self, "Помилка", f"Помилка при завантаженні
постів: {e}")

def view_posts_window(self):
    dialog = QDialog(self)
    dialog.setWindowTitle("Всі пости")
    layout = QVBoxLayout(dialog)

    self.postTable = QTableWidgetItem(dialog)
    layout.addWidget(self.postTable)

    self.update_post_button = QPushButton("Оновити пост", dialog)
    self.update_post_button.clicked.connect(self.update_post)
    layout.addWidget(self.update_post_button)

    self.delete_post_button = QPushButton("Видалити пост", dialog)
    self.delete_post_button.clicked.connect(self.delete_post)
    layout.addWidget(self.delete_post_button)

```

```
self.display_all_posts()

dialog.exec_()

def display_all_posts(self):
    query = """SELECT posts.id, posts.platform, posts.likes, posts.comments,
posts.audience, posts.days_since_publication, products.name
            FROM posts JOIN products ON posts.product_id = products.id"""
    rows = self.cursor.execute(query).fetchall()

    self.postTable.setRowCount(len(rows))
    self.postTable.setColumnCount(7)
    self.postTable.setHorizontalHeaderLabels(["ID", "Платформа", "Лайки",
"Коментарі", "Аудиторія", "Дні з моменту публікації", "Товар"])

    for i, row in enumerate(rows):
        for j, val in enumerate(row):
            self.postTable.setItem(i, j, QTableWidgetItem(str(val)))

def update_post(self):
    selected_items = self.postTable.selectedItems()
    if selected_items:
        post_id = int(selected_items[0].text())
        current_platform = selected_items[1].text()
        current_likes = selected_items[2].text()
        current_comments = selected_items[3].text()
        current_audience = selected_items[4].text()
        current_days = selected_items[5].text()
```

```
current_product = selected_items[6].text()

dialog = QDialog(self)
dialog.setWindowTitle("Оновити пост")
layout = QFormLayout(dialog)

platform_edit = QComboBox(dialog)
platform_edit.addItems(["Facebook", "Instagram", "Twitter", "Інше"])
platform_edit.setCurrentText(current_platform)
likes_edit = QLineEdit(dialog)
likes_edit.setText(current_likes)
comments_edit = QLineEdit(dialog)
comments_edit.setText(current_comments)
audience_edit = QLineEdit(dialog)
audience_edit.setText(current_audience)
days_edit = QLineEdit(dialog)
days_edit.setText(current_days)

product_combo = QComboBox(dialog)
product_combo.addItems([row[0] for row in self.cursor.execute("SELECT
name FROM products").fetchall()])
product_combo.setCurrentText(current_product)

layout.addRow("Платформа:", platform_edit)
layout.addRow("Лайки:", likes_edit)
layout.addRow("Коментарі:", comments_edit)
layout.addRow("Аудиторія:", audience_edit)
layout.addRow("Дні з моменту публікації:", days_edit)
```

```

layout.addRow("Товар:", product_combo)

update_button = QPushButton("Оновити", dialog)
update_button.clicked.connect(lambda: self.save_updated_post(post_id,
platform_edit.currentText(), likes_edit.text(),
                                comments_edit.text(),
audience_edit.text(), days_edit.text(),
                                product_combo.currentText(), dialog))

layout.addRow(update_button)

dialog.exec_()
else:
    QMessageBox.warning(self, "Помилка", "Будь ласка, оберіть пост для
оновлення.")

def save_updated_post(self, post_id, platform, likes, comments, audience, days,
product_name, dialog):
    try:
        product_id = self.cursor.execute("SELECT id FROM products WHERE
name=?", (product_name,)).fetchone()[0]
        self.cursor.execute("UPDATE posts SET platform=?, likes=?, comments=?,
audience=?, days_since_publication=?, product_id=? WHERE id=?",
                            (platform, int(likes), int(comments), int(audience), int(days),
product_id, post_id))
        self.conn.commit()
        QMessageBox.information(self, "Інформація", "Пост успішно оновлений!")
        dialog.accept()
        self.view_posts_window()

```

```

except Exception as e:
    QMessageBox.warning(self, "Помилка", f"Помилка при оновленні посту:
{e}")

def delete_post(self):
    selected_items = self.postTable.selectedItems()
    if selected_items:
        post_id = int(selected_items[0].text())
        confirm = QMessageBox.question(self, "Підтвердження", "Ви впевнені, що
хочете видалити цей пост?", QMessageBox.Yes | QMessageBox.No)
        if confirm == QMessageBox.Yes:
            try:
                self.cursor.execute("DELETE FROM posts WHERE id=?", (post_id,))
                self.conn.commit()
                QMessageBox.information(self, "Інформація", "Пост успішно
видалений!")
                self.view_posts_window()
            except Exception as e:
                QMessageBox.warning(self, "Помилка", f"Помилка при видаленні
посту: {e}")
            else:
                QMessageBox.warning(self, "Помилка", "Будь ласка, оберіть пост для
видалення.")

def validate_model(self, model, X, y):
    scores = cross_val_score(model, X, y, cv=5, scoring='neg_mean_squared_error')
    mean_score = np.mean(scores)
    QMessageBox.information(self, "Інформація", f"Середній MSE (перехресна

```

перевірка): {mean\_score:.2f}")

```

def train_model_window(self):
    dialog = QDialog(self)
    dialog.setWindowTitle("Навчити модель")
    layout = QFormLayout(dialog)

    method_combo = QComboBox(dialog)
    method_combo.addItems(["Лінійна регресія", "Дерево рішень", "Random
Forest"]) # Додано нові методи

    platform_combo = QComboBox(dialog)
    platform_combo.addItems(["Усі", "Facebook", "Instagram", "Twitter", "Інше"])

    layout.addRow("Метод навчання:", method_combo)
    layout.addRow("Соціальна мережа:", platform_combo)

    train_button = QPushButton("Навчити", dialog)
    train_button.clicked.connect(lambda:
self.train_model(method_combo.currentText(), platform_combo.currentText(),
dialog))
    layout.addRow(train_button)

    self.result_label = QLabel("", dialog)
    layout.addRow(self.result_label)

    dialog.exec_()

```

```

def train_model(self, method, platform, dialog):
    try:
        if platform == "Yci":
            query = """SELECT p.price, ps.likes, ps.comments, ps.audience,
ps.days_since_publication
                        FROM products p JOIN posts ps ON p.id = ps.product_id"""
            data = self.cursor.execute(query).fetchall()
        else:
            query = """SELECT p.price, ps.likes, ps.comments, ps.audience,
ps.days_since_publication
                        FROM products p JOIN posts ps ON p.id = ps.product_id WHERE
ps.platform = ?"""
            data = self.cursor.execute(query, (platform,)).fetchall()

        if not data:
            raise ValueError("Недостатньо даних для навчання моделі.")

        df = pd.DataFrame(data, columns=["price", "likes", "comments", "audience",
"days_since_publication"])
        X = df[["likes", "comments", "audience", "days_since_publication"]]
        y = df["price"]

        if method == "Лінійна регресія":
            model = LinearRegression()
        elif method == "Дерево рішень":
            model = DecisionTreeRegressor()
        elif method == "Random Forest":

```

```

        model = RandomForestRegressor()
    else:
        raise ValueError("Невідомий метод навчання.")

    # Виконання крос-валідації
    scores = cross_val_score(model, X, y, cv=5,
scoring='neg_mean_squared_error')
    mean_score = np.mean(scores)
    QMessageBox.information(self, "Інформація", f"Середній MSE
(перехресна перевірка): {mean_score:.2f}")

    model.fit(X, y)

    # Зберегти модель у файл, який враховує платформу
    platform_key = platform.replace(' ', '_').lower() if platform != "Yci" else "all"
    model_file_name = f"{method.replace(' ',
' ').lower()}_{platform_key}_model.pkl"
    with open(model_file_name, "wb") as model_file:
        pickle.dump(model, model_file)

    QMessageBox.information(self, "Інформація", f"Модель {method} для
платформи {platform} успішно навчена та збережена.")
    dialog.accept()
except Exception as e:
    QMessageBox.warning(self, "Помилка", f"Помилка при навчанні моделі:
{e}")

def predict_price_window(self):

```

```

dialog = QDialog(self)
dialog.setWindowTitle("Спрогнозувати ціну")
layout = QFormLayout(dialog)

method_combo = QComboBox(dialog)
method_combo.addItem("Лінійна регресія", "Дерево рішень", "Random
Forest"]) # Додано нові методи

platform_combo = QComboBox(dialog)
platform_combo.addItem("Yci", "Facebook", "Instagram", "Twitter", "Інше"])

category_combo = QComboBox(dialog)
category_combo.addItem([row[0] for row in self.cursor.execute("SELECT
DISTINCT category FROM products").fetchall()])

likes_edit = QLineEdit(dialog)
comments_edit = QLineEdit(dialog)
audience_edit = QLineEdit(dialog)
days_edit = QLineEdit(dialog)

layout.addRow("Метод прогнозування:", method_combo)
layout.addRow("Соціальна мережа:", platform_combo)
layout.addRow("Категорія товару:", category_combo)
layout.addRow("Лайки:", likes_edit)
layout.addRow("Коментарі:", comments_edit)
layout.addRow("Аудиторія:", audience_edit)
layout.addRow("Дні з моменту публікації:", days_edit)

```

```

    predict_button = QPushButton("Спрогнозувати", dialog)
    predict_button.clicked.connect(lambda:
self.predict_price(method_combo.currentText(), platform_combo.currentText(),
                    category_combo.currentText(),
likes_edit.text(),
                    comments_edit.text(), audience_edit.text(),
                    days_edit.text(), dialog))

    layout.addRow(predict_button)

    self.result_label = QLabel("", dialog)
    layout.addRow(self.result_label)

    dialog.exec_()

def predict_price(self, method, platform, category, likes, comments, audience, days,
dialog):
    try:
        # Підготовка даних для прогнозування
        data = pd.DataFrame([{
            "likes": int(likes),
            "comments": int(comments),
            "audience": int(audience),
            "days_since_publication": int(days)
        }])

        # Формування імені файлу моделі з урахуванням платформи
        platform_key = platform.replace(' ', '_').lower() if platform != "Усі" else "all"
        model_file_name = f'{method.replace(' ',

```

```

'_.lower()}_{platform_key}_model.pkl"
    with open(model_file_name, "rb") as model_file:
        model = pickle.load(model_file)

# Прогнозування ціни за допомогою завантаженої моделі
predicted_price = model.predict(data)[0]

# Пошук середньої ціни товарів у вибраній категорії
query = "SELECT AVG(price) FROM products WHERE category = ?"
average_price = self.cursor.execute(query, (category,)).fetchone()[0]

self.result_label.setText(f'Прогнозована ціна: {predicted_price:.2f} грн
(Середня ціна в категорії: {average_price:.2f} грн)')
except Exception as e:
    self.result_label.setText(f'Сталася помилка під час прогнозування ціни:
{e}')

def display_data(self, data):
    self.tableWidget.setRowCount(len(data))
    self.tableWidget.setColumnCount(len(data.columns))
    self.tableWidget.setHorizontalHeaderLabels(data.columns)

    for i in range(len(data)):
        for j in range(len(data.columns)):
            self.tableWidget.setItem(i, j, QTableWidgetItem(str(data.iloc[i, j])))

def read_products_from_excel(file_path):

```

```

try:
    data = pd.read_excel(file_path)
    expected_columns = ["Назва товару", "Ціна", "Категорія"]
    if not all(col in data.columns for col in expected_columns):
        raise ValueError(f"Очікувані стовпці відсутні в файлі: {'',
'.join(expected_columns)}")
    return data
except Exception as e:
    print("Помилка під час зчитування даних з Excel файлу:", e)
    return None

def read_posts_from_excel(file_path):
    try:
        data = pd.read_excel(file_path)
        expected_columns = ["Платформа", "Лайки", "Коментарі", "Аудиторія", "Дні
з моменту публікації", "Товар"]
        if not all(col in data.columns for col in expected_columns):
            raise ValueError(f"Очікувані стовпці відсутні в файлі: {'',
'.join(expected_columns)}")
        return data
    except Exception as e:
        print("Помилка під час зчитування даних з Excel файлу:", e)
        return None

def get_average_price(data):
    try:

```

```
    average_price = data["Середня ціна"].mean()
    return average_price
except Exception as e:
    print("Помилка під час отримання середньої ціни:", e)
    return None
```

```
def generate_coefficients(data):
    try:
        average_price = get_average_price(data)
        beta0 = average_price
        beta1 = np.random.uniform(0, 0.05)
        beta2 = np.random.uniform(0, 0.0001)
        beta3 = np.random.uniform(0, 0.1)
        beta4 = np.random.uniform(0, 0.00001)
        return beta0, beta1, beta2, beta3, beta4
    except Exception as e:
        print("Помилка під час генерації коефіцієнтів:", e)
        return None
```

```
def generate_prices(data, coefficients):
    try:
        beta0, beta1, beta2, beta3, beta4 = coefficients
        print("Data for price generation:", data)
        print("Coefficients:", coefficients)

        prices = []
```

```
for index, row in data.iterrows():
    price = beta0 + beta1 * row["Лайки"] + beta2 * row["Коментарі"] + beta3 *
row["Аудиторія"] + beta4 * row[
    "Дні з початку публікації"]
    prices.append(price)

return prices
except Exception as e:
    print("Помилка під час генерації цін:", e)
    return None

if __name__ == '__main__':
    app = QApplication([])
    ex = PricePredictionApp()
    ex.show()
    sys.exit(app.exec_())
```