

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут  
«Інститут інформаційних технологій в економіці»**

**Кафедра математичного моделювання та статистики**

**Освітньо-професійна програма «Економічна кібернетика і Дата Сайнс»**

Галузь знань 05 «Соціальні та поведінкові науки»  
Спеціальність 051 «Економіка»

Форма навчання: очна (денна)

**КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

на тему **«Data-Driven підхід в обґрунтуванні вибору об'єктів приватного  
житлового інвестування»**  
(назва теми)

здобувача Овчинникова Ярослава Сергійовича  
(ПІБ, підпис)

Науковий керівник: кандидат економічних наук, доцент Ольга КАТУНІНА  
(науковий ступінь, учене звання, ПІБ)

\_\_\_\_\_  
(підпис)

**Робота допущена до захисту перед екзаменаційною  
комісією з атестації здобувачів вищої освіти (ЕК)**

Завідувач кафедри: к. ф.-м. н., професор  
Галина ВЕЛИКОІВАНЕНКО

\_\_\_\_\_  
(підпис)

**Київ 2024**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут  
«Інститут інформаційних технологій в економіці»**

**Кафедра математичного моделювання та статистики**

**Освітньо-професійна програма «Економічна кібернетика і Дата Сайнс»**

Галузь знань 05 «Соціальні та поведінкові науки»  
Спеціальність 051 «Економіка»

**ПОГОДЖЕНО**

Керівник проектної групи (гарант)  
освітньо-професійної програми  
Олена ПІСКУНОВА

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
2024 р.

**ЗАТВЕРДЖУЮ**

Завідувач кафедри математичного  
моделювання та статистики  
Галина ВЕЛИКОІВАНЕНКО

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
2024 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**здобувача вищої освіти**

Овчинникова Ярослава Сергійовича

(прізвище, ім'я, по батькові)

очної (денної)

**форми навчання**

очної (денної), заочної, дистанційної

на підготовку кваліфікаційної магістерської роботи

**на тему**

«Data-Driven підхід в обґрунтуванні вибору об'єктів приватного житлового  
інвестування»

Тему затверджено наказом ректора Університету від «19» вересня 2024 р. № 1610 / ст

**Кваліфікаційна магістерська робота виконується на матеріалах  
офіційних сайтів з продажу об'єктів нерухомості**

## План кваліфікаційної магістерської роботи

|          |                                                                                                                                    |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| Розділ 1 | Теоретичні основи використання Data Driven підходу при виборі об'єктів приватного житлового інвестування                           |
| Розділ 2 | Аналіз методів автоматизованого машинного та глибокого навчання для оцінювання вартості об'єктів приватного житлового інвестування |
| Розділ 3 | Практична реалізація Data Driven підходу та модельного інструментарію для вибору об'єктів житлового інвестування                   |

|                                                      |                                                                                                                                                |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| Об'єкт дослідження:                                  | Вибір об'єктів приватного житлового інвестування                                                                                               |
| Предмет дослідження:                                 | Методи машинного та глибокого навчання для вибору об'єктів приватного житлового інвестування                                                   |
| Мета виконання кваліфікаційної магістерської роботи: | Розробка моделей машинного та глибокого навчання для аналізу та прогнозування ціни на житлові об'єкти з урахування даних орієнтованого підходу |

**Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:**

**У розділі 1:** 1) дослідити сучасний стан сектору нерухомості України; 2) проаналізувати існуючі методи та підходи в прогнозуванні ціни на об'єкти нерухомості та обґрунтуванні вибору об'єктів нерухомості для інвестування; 3) розглянути основи даних орієнтованого підходу для прийняття рішень щодо об'єктів приватного житлового інвестування

**У розділі 2:** опрацювати алгоритми машинного та глибокого навчання для визначення ціни на об'єкти нерухомості

**У розділі 3:** 5) розробити комплекс моделей для оцінки вартості об'єктів житлової нерухомості; 6) проаналізувати отримані результати моделей оцінки вартості об'єктів нерухомості

Завдання підготував  
науковий керівник

\_\_\_\_\_  
(підпис)

Ольга КАТУНІНА

(ім'я, прізвище)

«\_\_» \_\_\_\_\_ 2024 р.

Завдання одержав  
здобувач

\_\_\_\_\_  
(підпис)

Ярослав ОВЧИННИКОВ

(ім'я, прізвище)

«\_\_» \_\_\_\_\_ 2024 р.

## Реферат

Кваліфікаційна магістерська робота містить 91 сторінки, 14 таблиць, 38 рисунків, список використаних джерел з 80 найменувань, додатки.

### **«Data-Driven підхід в обґрунтуванні вибору об'єктів приватного житлового інвестування»**

*Об'єктом дослідження* кваліфікаційної магістерської роботи є вибір об'єктів приватного житлового інвестування.

*Предметом дослідження* є методи машинного та глибокого навчання для прогнозування ціни об'єктів нерухомості та вибору об'єктів приватного житлового інвестування.

*Мета та завдання дослідження.* Розробка моделей машинного та глибокого навчання для аналізу та прогнозування ціни на житлові об'єкти з урахування даних орієнтованого підходу при виборі об'єктів для інвестування.

Відповідно до поставленої мети визначені такі *завдання*:

- Дослідити сучасний стан сектору нерухомості України.
- Проаналізувати існуючі методи та підходи в прогнозуванні ціни на об'єкти нерухомості та обґрунтуванні вибору об'єктів нерухомості для інвестування.
- Розглянути основи даних орієнтованого підходу для прийняття рішень щодо об'єктів приватного житлового інвестування.
- Опрацювати алгоритми машинного та глибокого навчання для визначення ціни на об'єкти нерухомості.
- Розробити комплекс моделей для оцінки вартості об'єктів житлової нерухомості.
- Проаналізувати отримані результати моделей оцінки вартості об'єктів нерухомості.

*Теоретична, методична та практична значущість отриманих результатів.*

Теоретична, методична та практична значущість отриманих результатів. Кваліфікаційна робота має теоретичну значущість оскільки поглиблює розуміння застосування технологій машинного навчання та важливості даних орієнтованого підходу для аналізу сучасного стану та тенденцій на ринку нерухомості.

Дослідження має методичну значущість, адже розроблені методики та алгоритми на основі нейронних мереж, можуть бути взяті як базис для подальших наукових досліджень та вдосконалень методологій, щодо прогнозування цін та виборі об'єктів у сфері житлового інвестування.

Практична значущість роботи полягає у створенні автоматизованої системи оцінювання вартості нерухомості, яка дозволяє різним зацікавленим сторонам, таким як інвестори,

забудовники, ріелтори та потенційні покупці, приймати обґрунтовані рішення щодо купівлі, продажу чи інвестування в житлові об'єкти. Крім того, дослідження передбачає аналіз та ідентифікацію ключових факторів, що впливають на ціноутворення на ринку житла, що дозволяє розробити стратегії для прийняття більш виважених рішень у сфері нерухомості.

Рік виконання кваліфікаційної магістерської роботи – 2024.

Рік захисту роботи – 2024.

*Ключові слова: ціна, об'єкти приватного житлового інвестування, автоматизоване машинне навчання, глибоке навчання, нейронні мережі.*

В і д г у к  
про кваліфікаційну магістерську роботу  
здобувача освітньо-професійної програми «Економічна кібернетика і Дата  
Сайнс»  
навчально-наукового інституту  
«Інститут інформаційних технологій в економіці»  
Овчинникова Ярослава Сергійовича  
на тему "Data-Driven підхід в обґрунтуванні вибору об'єктів приватного  
житлового інвестування"

1. Актуальність теми: Вибір об'єктів для інвестування в приватному житловому секторі є складним завданням, що потребує ретельного аналізу та врахування багатьох чинників. Застосування Data-Driven підходу, що базується на використанні великих даних та аналітичних інструментах машинного навчання, для отримання об'єктивної інформації та прогнозування майбутніх тенденцій, може значно підвищити якість прийняття таких рішень та зменшити інвестиційні ризики.

2. Позитивні риси кваліфікаційної магістерської роботи: У дослідженні було застосовано передові методи машинного та глибокого навчання, для прогнозування цін та обґрунтування вибору об'єктів житлової нерухомості. Зокрема, використовувалися нейронні мережі, які є потужними інструментами для моделювання складних нелінійних залежностей між вхідними даними та цільовими змінними. Крім того, було впроваджено автоматизоване машинне навчання, що дозволило значно підвищити ефективність та швидкість проведення досліджень. Окрім того, в роботі були застосовані методи обробки текстових даних, що забезпечило більш повне та точне представлення інформації, необхідної для прийняття обґрунтованих рішень щодо інвестицій у нерухомість. Загалом, це дослідження продемонструвало ефективне поєднання сучасних технологій штучного інтелекту для вирішення складної задачі прогнозування цін на нерухомість.

3. Наявність самостійних розробок автора: Автором було здійснено аналіз та обробку вхідних даних щодо об'єктів нерухомості. Побудовано комплекс моделей машинного навчання на основі архітектури нейронних мереж для підвищення якості прийняття інвестиційних рішень. Перевірено якість та значущість отриманих результатів, використовуючи різні метрики оцінювання.

4. Цінність теоретичних висновків та практичних рекомендацій: Розроблені в рамках дослідження моделі машинного навчання, засновані на архітектурі нейронних мереж, надають інвесторам потужний інструмент для автоматизованої оцінки вартості об'єктів житлової нерухомості. Ці моделі можуть слугувати підґрунтям для прийняття зважених рішень стосовно інвестування в нерухомість шляхом купівлі або продажу житлових об'єктів. Результати цього дослідження були представлені та успішно захищені на науковій конференції, присвяченій

модельовання економіки та дослідженню методів прогнозування.

5. Наявність недоліків: У контексті розглянутих у роботі моделей машинного навчання для обґрунтування вибору об'єктів нерухомості, було б доцільно більш ретельно інтегрувати моделі економічної динаміки. Такий підхід дозволив би краще врахувати як довгострокові тенденції, так і короткострокові коливання, а також структурні зміни, що відбуваються на ринку нерухомості.

6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК: Кваліфікаційна магістерська робота студента Овчинникова Ярослава Сергійовича є актуальною, та демонструє автором глибоке розуміння фундаментальних знань у сфері технологій машинного навчання. На мою думку, ця робота заслуговує на те, щоб бути представленою для захисту перед екзаменаційною комісією, та гідна високої оцінки.

Науковий керівник кандидат економічних наук  
(посада, учене звання, науковий ступінь)

\_\_\_\_\_  
(підпис)  
“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

Ольга КАТУНІНА  
(ім'я, ПРІЗВИЩЕ)

## РЕЦЕНЗІЯ

на кваліфікаційну магістерську роботу  
здобувача навчально-наукового інституту "Інститут інформаційних технологій в  
економіці" Київського національного економічного університету імені Вадима Гетьмана  
освітньо-професійної програми "Економічна кібернетика і Дата Сайнс"  
Овчинникова Ярослава Сергійовича  
на тему "Data-Driven підхід в обґрунтуванні вибору об'єктів приватного житлового  
інвестування"

**Актуальність теми кваліфікаційної магістерської роботи і доцільність її розроблення.** Проблема ефективного інвестування у приватний житловий сектор є актуальною, оскільки вибір об'єктів для інвестицій вимагає глибокого аналізу та урахування багатьох факторів. Застосування Data-Driven підходу та математичного моделювання може значно підвищити якість прийняття рішень, знизити ризики та забезпечити систематичність інвестиційного процесу.

**Якість проведеного дослідження.** У роботі проведено аналіз сучасного стану ринку нерухомості в Україні, виокремлено важливість застосування Data-Driven підходу для обґрунтування вибору житлових об'єктів для інвестицій. Застосовано сучасні технології машинного та глибокого навчання для побудови комплексу моделей на базі нейронних мереж, розроблено математичну модель, яка демонструє високу точність та надійність для прогнозування вартості потенційних об'єктів для інвестицій в нерухомість.

**Позитивні риси кваліфікаційної магістерської роботи.** У дослідженні ефективно використано сучасні технології машинного та глибокого навчання на базі нейронних мереж для прогнозування вартості об'єктів житлової нерухомості. Застосовано автоматизоване машинне навчання, що значно підвищило продуктивність дослідження. Препроцесинг здійснено, зокрема для неструктурованих даних, практично реалізовано методи обробки текстової інформації. Побудовано комплекс моделей для визначення ціни на житло, що забезпечило поглиблений ступінь обґрунтування рішень щодо інвестицій в нерухомість.

**Зауваження.** Було б корисно дослідити моделі економічної динаміки, які базуються на технологіях нейромереж для врахування часових тенденцій та низки супутніх процесів, що впливають на ринок нерухомості.

**Практична значимість висновків і рекомендацій.** Побудовані моделі автоматизованої оцінки вартості об'єктів нерухомості можуть бути використані інвесторами для ухвалення рішень щодо купівлі чи продажу житлових об'єктів, визначення цінових параметрів житла та сегментування пропозицій на ринку нерухомості України.

**Загальний висновок рецензента і оцінка кваліфікаційної магістерської роботи.** У кваліфікаційній магістерській роботі Овчинникова Ярослава Сергійовича на тему "Data-Driven підхід в обґрунтуванні вибору об'єктів приватного житлового інвестування" продемонстровано глибокі знання в галузі сучасних технологій математичного моделювання та машинного навчання, одержано практичні результати, сформульовано логічні висновки та рекомендації. Проведене дослідження є достатньо повним та обґрунтованим, виконано на високому рівні та відповідає вимогам до кваліфікаційної магістерської роботи. Вважаю, що робота Овчинникова Я.С. може бути допущена до захисту в ДЕК і заслуговує високої оцінки.

### Рецензент

Місце роботи та посада рецензента

Науковий ступінь, учене звання (за наявності) \_\_\_\_\_  
(підпис, ПІБ)

Підпис засвідчую:

\_\_\_\_\_ (посада, підпис)

Місце печатки організації, де працює рецензент

## ЗМІСТ

|                                                                                                                                                                           |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ВСТУП.....</b>                                                                                                                                                         | <b>3</b>   |
| <b>РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ DATA DRIVEN ПІДХОДУ ПРИ ВИБОРІ ОБ'ЄКТІВ ПРИВАТНОГО ЖИТЛОВОГО ІНВЕСТИВАННЯ.....</b>                                             | <b>6</b>   |
| 1.1 Значення прогнозування ціни на нерухомість для економіки та учасників ринку нерухомості при виборі потенційних об'єктів житлових інвестицій .....                     | 6          |
| 1.2 Огляд існуючих методів та підходів до прогнозування цін на нерухомість, їх переваги та недоліки.....                                                                  | 11         |
| 1.3 Теоретичні основи Data Driven підходу та його роль у прийнятті рішень в економіці .....                                                                               | 17         |
| <b>РОЗДІЛ 2 АНАЛІЗ МЕТОДІВ АВТОМАТИЗОВАНОГО МАШИННОГО ТА ГЛИБОКОГО НАВЧАННЯ ДЛЯ ОЦІНЮВАННЯ ВАРТОСТІ ОБ'ЄКТІВ ПРИВАТНОГО ЖИТЛОВОГО ІНВЕСТИВАННЯ.....</b>                   | <b>29</b>  |
| 2.1 Інструментарій обґрунтування рішень при виборі приватних житлових об'єктів на базі аналітики даних та машинно-навчальних технологій моделювання .....                 | 29         |
| 2.2 Використання нейромережевого моделювання для оцінювання вартості житла .....                                                                                          | 33         |
| 2.3 Автоматизоване машинне навчання для побудови нейронних мереж .....                                                                                                    | 47         |
| <b>РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ DATA DRIVEN ПІДХОДУ ТА МОДЕЛЬНОГО ІНСТРУМЕНТАРІЮ ДЛЯ ВИБОРУ ОБ'ЄКТІВ ЖИТЛОВОГО ІНВЕСТИВАННЯ .....</b>                                    | <b>52</b>  |
| 3.1 Результати препроцесингу структурованих та неструктурованих даних для моделювання ціни на об'єкти нерухомості .....                                                   | 52         |
| 3.2 Аналіз результатів нейромережевого моделювання та прогнозування ціни на житлову нерухомість.....                                                                      | 61         |
| 3.3 Аналіз результатів предиктивного моделювання вартості житла на базі алгоритмів оптимізації гіперпараметрів та інструментарію автоматизованого машинного навчання..... | 80         |
| <b>ВИСНОВКИ.....</b>                                                                                                                                                      | <b>89</b>  |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>                                                                                                                                    | <b>92</b>  |
| <b>ДОДАТКИ .....</b>                                                                                                                                                      | <b>100</b> |

## ВСТУП

*Актуальність теми.* Ринок нерухомості є надзвичайно динамічним та мінливим. Ціни на нерухомість можуть значно коливатися залежно від економічної ситуації в країні та регіоні. Для інвесторів, які прагнуть максимізувати свої прибутки та мінімізувати ризики, критично важливо мати доступ до актуальних та достовірних даних, які можуть допомогти їм приймати обґрунтовані рішення щодо вибору об'єктів для інвестування.

Традиційні методи оцінки та аналізу нерухомості часто базуються на суб'єктивних факторах, таких як досвід та інтуїція експертів з нерухомості. Хоча ці методи мають свою цінність, вони можуть бути недостатньо точними та не враховувати всі релевантні дані та тенденції на ринку. Data Driven підхід, який передбачає використання великих обсягів даних та аналітичних інструментів, може забезпечити більш об'єктивний та точний аналіз для прийняття обґрунтованих рішень та прогнозування результатів.

Використання великих наборів даних, та ретельний аналіз ринку нерухомості, дозволяє розуміти ринкові тенденції, ризики, пов'язані з різними об'єктами нерухомості. Моделі машинного навчання, які працюють на основі великих обсягів даних, можуть використовувати ці відомості для виявлення складних закономірностей та тенденцій. Використання машинного навчання на основі великих даних, дозволяє вибрати найбільш привабливі, об'єкти для інвестицій та забезпечує вищу ймовірність отримання прибуткового результату.

Використання Data Driven в сфері житлового інвестування допомагає підвищити ефективність управління інвестиційними портфелями. Завдяки використанню великих даних, інвестори і фінансові аналітики можуть більш точно оцінювати прибутковість різних активів і, відповідно, більш точно управляти цими активами. Також, актуальність дипломної роботи підкреслюється тим, що використання моделей машинного навчання на основі великих даних, дозволяє

інвесторам швидко реагувати на зміни на ринку нерухомості, мінімізуючи таким чином ризики та максимізуючи можливості для прибутку.

*Аналіз останніх публікацій та досліджень.* Для отримання теоретичних знань про ринок нерухомості в Україні було досліджено роботи С. Шибріна, В. Запотоцька, В. Міщенко. Практичною реалізацією та застосуванням методів моделювання ринку нерухомості займалися: Н. Максишко, В. Шаповалова, В. Григорків, О. Ярошенко. Теоретичні основи дата орієнтованого підходу розкриваються в роботах С. Брантона, Т. Чавеса, К. О'хара. Аспекти теорії прийняття рішень досліджені С. Братушка, С. Новак, О. Горда.

*Мета і завдання дослідження.* Розробка моделей машинного та глибокого навчання для аналізу та прогнозування ціни на житлові об'єкти з урахування дата орієнтованого підходу.

Досягнення поставленої мети обумовлює необхідність вирішення таких завдань:

1. Дослідити сучасний стан сектору нерухомості України.
2. Проаналізувати існуючі методи та підходи в прогнозуванні та обґрунтуванні вибору об'єктів нерухомості.
3. Розглянути основи дата орієнтованого підходу для прийняття рішень щодо об'єктів приватного житлового інвестування.
4. Опрацювати алгоритми машинного та глибокого навчання для визначення ціни на об'єкти нерухомості.
5. Розробити комплекс моделей для оцінки вартості об'єктів житлової нерухомості.
6. Проаналізувати отримані результати моделей оцінки вартості об'єктів нерухомості.

*Об'єктом дослідження* кваліфікаційної магістерської роботи є вибір об'єктів приватного житлового інвестування.

*Предметом дослідження* є методи машинного та глибокого навчання для вибору об'єктів приватного житлового інвестування.

Методами дослідження даної роботи є передові методи машинного навчання, такі як автоматизоване машинне навчання, глибоке навчання, зокрема нейронні мережі, а також спеціалізовані підходи до аналізу та обробки текстових даних. Для ефективної роботи з великими обсягами інформації застосовуються сучасні методи обробки Big Data.

*Теоретична, методична та практична значущість отриманих результатів.*

Кваліфікаційна робота має теоретичну значущість оскільки поглиблює розуміння застосування технологій машинного навчання та важливості дата орієнтованого підходу для аналізу сучасного стану та тенденцій на ринку нерухомості.

Дослідження має методичну значущість, адже розроблені методики та алгоритми на основі нейронних мереж, можуть бути взяті як базис для подальших наукових досліджень та вдосконалень методологій, щодо прогнозування цін та виборі об'єктів у сфері житлового інвестування.

Практична значущість роботи полягає у створенні автоматизованої системи оцінювання вартості нерухомості, яка дозволяє різним зацікавленим сторонам, таким як інвестори, забудовники, ріелтори та потенційні покупці, приймати обґрунтовані рішення щодо купівлі, продажу чи інвестування в житлові об'єкти. Крім того, дослідження передбачає аналіз та ідентифікацію ключових факторів, що впливають на ціноутворення на ринку житла, що дозволяє розробити стратегії для прийняття більш виважених рішень у сфері нерухомості.

*Інформаційна база дослідження.* Інформаційною базою роботи є праці вітчизняних та зарубіжних науковців, наукові статті, статистичні дані з спеціалізованих інтернет-джерел та науково-методична література.

*Структура роботи.* Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків.

## РОЗДІЛ 1

### ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ DATA DRIVEN ПІДХОДУ ПРИ ВИБОРІ ОБ'ЄКТІВ ПРИВАТНОГО ЖИТЛОВОГО ІНВЕСТИВАННЯ

#### **1.1 Значення прогнозування ціни на нерухомість для економіки та учасників ринку нерухомості при виборі потенційних об'єктів житлових інвестицій**

Наявність житла є однією з найважливіших базових потреб у житті людини, поряд з іншими основними потребами, такими як їжа та вода. Попит на власне чи орендоване житло швидко зростає протягом останніх років через повномасштабне вторгнення Росії в Україну. Хоча є люди, які розглядають свій будинок як інвестицію та актив, більшість людей у всьому світі купують чи орендують будинок для проживання.

Безсумнівно, житловий сектор позитивно впливає на валюту країни, що є важливим показником для національної економіки. Домовласники будуть купувати товари, такі як меблі та побутова техніка для свого будинку, а забудовники чи підрядники будуть купувати сировину для будівництва будинків, щоб задовольнити попит на житло, що є прикладом ефекту економічної хвилі [1], створеного новою пропозицією житла. Тим часом споживачі мають капітал для здійснення великих інвестицій, а стан будівельної галузі, чи то процвітаючий чи занепадаючий, можна побачити через високий рівень пропозиції чи попиту на житло в країні. Однак мати власне житло стає все більш недоступним активом, оскільки спостерігається поступове зростання цін на ринку нерухомості в багатьох країнах.

В умовах високої інфляції, нестабільності на ринках та недосконалої пенсійної системи нерухомість розглядається як один з найкращих способів зберегти заощадження. Інвестори звертаються до нерухомості, коли відчувають

невпевненість щодо фінансових ринків, оскільки це дозволяє їм диверсифікувати ризики [2].

Нерухомість, як предмет операцій на ринку нерухомості, має ринкову ціну, а отже, залежно від операції з нею характеризується певною вартістю. В умовах ринкової економіки ціна формується під впливом різноманітних факторів та механізмів, а отже набуває форму ринкової вартості. В оціночній діяльності під ринковою вартістю, за яку можливе відчуження об'єкта оцінки на ринку подібного майна на дату оцінки за угодою, укладеною між покупцем та продавцем, після проведення відповідного маркетингу за умови, що кожна із сторін діяла із знанням справи, розсудливо і без примусу [3]. Отже, ринкова вартість є об'єктивною та відображає взаємовідносини покупця та продавця в реальних економічних умовах ринку. Фактором формування вартості об'єкту на будь якому ринку є взаємодія попиту та пропозиції. Попит на ринку нерухомості можна визначити як кількість об'єктів нерухомості та прав на них, які покупець бажає придбати за сформованою ціною за певний проміжок часу [4]. Відповідно під пропозицією на ринку нерухомості слід розуміти кількість об'єктів нерухомості, яку власник бажає продати за певну ціну та проміжок часу.

Відображенням кон'юнктури на ринку нерухомості, тобто співвідношення попиту та пропозиції, є вартість на нерухоме майно, яка виражається його ціною. Так, за умов великої пропозиції, коли ринок перенасичений існуючою та новою нерухомістю і відповідно характеризується низькою ціною, попит невеликий, що характеризує ринок покупця. За такої ринкової ситуації попит має низьку еластичність за ціною, джерелом якої є відсутність взаємозамінності різних видів нерухомості. Ринок продавця має місце за умов зростання попиту на нову нерухомість й відповідно ціни, в результаті з'являється надлишок будівельних потужностей і скороченням будівельної діяльності. В даному випадку пропозиція характеризується низькою еластичністю за ціною як наслідок тривалості процесу будівництва. В результаті низької еластичності попиту та пропозиції ускладнюється процес їх взаємопристосування та встановлення ринкової рівноваги на ринку нерухомості.

Ринок нерухомості, як специфічний сектор економіки, характеризується рядом особливостей, що пов'язані з ознаками самої нерухомості, а саме: фізичними властивостями об'єкта нерухомості, та економічними властивостями об'єкта нерухомості [5]. Під впливом економічних факторів формується середній рівень цін угод. До економічних факторів належать: економічний рівень розвитку країни, темпи зростання ВВП, грошово-кредитна політика, темпи інфляції, коливання ділової активності та інші. Економічними факторами, що визначають попит, є: зайнятість населення, рівень середньої заробітної плати та доходів населення, рівень цін, наявність джерел фінансування, доступність та умови кредиту на купівлю житла. Фізичними факторами є: характеристика місця розташування та об'єкта. Враховується рівень розвитку конкретного регіону, міста, характер сусіднього оточення або престижність району, відстань від центру, ступінь розвитку інфраструктури і транспортного сполучення (наприклад, близькість зупинок транспорту); характеристики власне ділянки: розмір, форма, площа, під'їзди, благоустрій, загальний вид; характеристики будівель та споруд: тип будівлі, рік побудови, стиль, стан будівлі.

Повномасштабне вторгнення Росії в Україну стало вагомим економічним чинником, що суттєво вплинув на формування цін на ринку нерухомості. Бойові дії спричинили значні потрясіння на ринку нерухомості, змінивши тенденції та пріоритети як для покупців, так і для інвесторів. Бойові дії змусили багатьох людей переміщуватися всередині країни, що призвело до зростання попиту на житло у відносно безпечних західних регіонах, тоді як на сході ринок нерухомості практично заморозився. Ця внутрішня міграція створила нові виклики для будівельників, які мають задовольнити підвищений попит у певних областях. Крім того, нестабільна економічна ситуація, спричинена війною, ускладнила реалізацію нових будівельних проектів. Зростання цін на будматеріали та інфляція збільшили витрати на будівництво, змушуючи забудовників переглядати свої плани та бюджети. Водночас руйнування житлової інфраструктури в зонах бойових дій створило потребу у відновленні та реконструкції, відкриваючи нові можливості для інвесторів.

Незважаючи на труднощі, з якими стикається галузь нерухомості через війну, деякі учасники ринку розглядають ситуацію як шанс для інвестицій та розвитку. Очікується, що після закінчення конфлікту відбудеться значне зростання попиту на житло [6]. Після завершення війни український ринок нерухомості може стати більш привабливим для іноземних інвесторів. Проведення реформ та відбудова зруйнованих міст зроблять Україну цікавим напрямком для вкладення коштів. Вирішальним фактором тут стане інтеграція країни до Європейського Союзу, що відкриє шлях до залучення закордонних інвестицій та фінансування для відновлення пошкодженої інфраструктури. Очікується, що в 2025 року ринок нерухомості в Україні поступово відновиться, особливо в регіонах, які постраждали від бойових дій [7]. Однак динаміка цін на житло в західних областях значною мірою залежатиме від розвитку подій на фронті та можливості мешканців зі сходу повернутися до рідних міст після завершення війни. Якщо ситуація стабілізується і внутрішні переселенці зможуть виїхати із західних регіонів, попит на житло там може знизитися, що потенційно призведе до падіння цін.

Загалом, відбудова та європейська інтеграція відкриють нові перспективи для інвесторів на українському ринку нерухомості після закінчення війни. Але ціни в різних регіонах країни розвиватимуться по-різному, залежно від безпекової ситуації та міграційних потоків населення [8].

Ситуація на ринку нерухомості в Україні демонструє значні регіональні відмінності в цінах на новобудови. У західних областях, таких як Львівська чи Закарпатська, спостерігається різке підвищення вартості квадратного метра житла. Це пов'язано з високим попитом, який виник внаслідок масового переміщення людей із небезпечних східних регіонів. Наприклад, у Львові ціни на новобудови зросли на 20-30% порівняно з довоєнним періодом. У центральних регіонах країни ціни також підвищилися, проте не так стрімко, оскільки економічна ситуація та попит тут більш стабільні. У Києві вартість новобудов залишається відносно високою, хоча темпи зростання сповільнилися через скорочення платоспроможного попиту. Натомість на сході України, де тривають активні бойові дії, ціни впали.

Незважаючи на економічні чинники описані раніше, прогнозування цін на нерухомість відіграє вирішальну роль для економіки та учасників ринку нерухомості з кількох причин. По-перше, це допомагає приймати більш обґрунтовані інвестиційні рішення. Інвестори, які можуть передбачити майбутні ціни на нерухомість, матимуть перевагу над конкурентами та зможуть максимізувати свої прибутки. Точні прогнози допоможуть їм визначити найбільш привабливі об'єкти для інвестицій та уникнути ризиків, пов'язаних з переоціненими активами. По-друге, прогнозування цін на нерухомість є критично важливим для забудовників та девелоперів. Вони повинні ретельно оцінювати майбутній попит та ціни, перш ніж розпочинати нові проєкти. Якщо ціни на нерухомість зростатимуть повільніше, ніж очікувалося, це може зробити проєкти неприбутковими або навіть збитковими. З іншого боку, якщо ціни зростатимуть швидше, ніж прогнозувалося, девелопери можуть втратити потенційні прибутки через недостатню пропозицію.

Для покупців нерухомості прогнози вартості об'єктів нерухомості також мають велике значення. Вони допомагають визначити найкращий час для покупки. Крім того, прогнозування цін на нерухомість відіграє важливу роль для банків, які надають іпотечні кредити. Точні прогнози допомагають їм оцінювати ризики та визначати відповідні відсоткові ставки. Якщо очікується зростання цін, фінансові установи можуть бути більш схильними до надання іпотечних кредитів, оскільки ризик дефолту буде нижчим.

Незважаючи на складність прогнозування у воєнний час, точні оцінки майбутніх цін на нерухомість мають величезне значення для всіх учасників ринку [9]. Для України прогнозування цін на нерухомість набуває особливої важливості в умовах післявоєнної відбудови та відновлення економіки. Точні прогнози допоможуть залучити інвестиції в будівельний сектор, забезпечити доступне житло для постраждалих від війни та сприятимуть ефективному використанню ресурсів для відновлення зруйнованої інфраструктури.

## **1.2 Огляд існуючих методів та підходів до прогнозування цін на нерухомість, їх переваги та недоліки**

Точне прогнозування цін на житло є важливим завданням у сфері нерухомості, що дозволяє власникам будинків, покупцям та інвесторам приймати обґрунтовані рішення. Протягом останніх років алгоритми машинного навчання стали потужними інструментами для аналізу складних моделей даних та здійснення точних прогнозів.

Ранні роботи вивчали ціни на житло з використанням математичних та статистичних підходів. Гедоністична регресія - дуже популярний метод у ранніх роботах з визначення впливових змінних цін на житло, яка виявляє вплив заданих змінних на загальну ціну нерухомості та використовується для оцінки ринкових товарів за їхніми корисними характеристиками [10].

В останні десятиліття, з появою комп'ютерних моделей, багато досліджень застосовували передові моделі для аналізу ринку нерухомості, виявлення найвпливовіших змінних та пропозицій відповідної ціни. Було застосовано багато алгоритмів та підходів машинного навчання для оцінки їх успіху в отриманні точних прогнозів. Основні алгоритми та роботи, зосереджуються на навчанні загальних статистичних та машинних моделей навчання, таких як нечітка логіка, нейронні мережі, лінійна регресія, випадковий ліс, генетичні алгоритми. В останні роки, роботи використовували більш складні алгоритми такі як нейронні мережі, глибоке навчання та градієнтний бустинг для вирішення проблем точного прогнозування [11, 12, 13].

Алгоритми машинного навчання можуть ефективно виявляти приховані закономірності та взаємозв'язки у великих наборах даних, що містять безліч факторів, які впливають на ціни на нерухомість. Ці фактори можуть включати місце розташування, розмір житла, кількість кімнат, наявність додаткових зручностей, а також демографічні та економічні показники регіону.

Для прогнозування ціни, можна використовувати дані з попередніх продажів, крім того, можна використовувати машинне навчання для створення цього прогнозу.

Завищення та заниження цін на ринку житла завжди були проблемами, і немає достатньо ефективних методів їх виявлення. Первинну оцінку можна отримати за допомогою широких показників, таких як співвідношення цін на житло/нерухомість до орендної плати. Однак для вирішення цього питання потрібна ретельна оцінка та прийняття рішень, саме тут і допомагає машинне навчання. Надаючи моделі машинного навчання тисячі точок даних для навчання, можна створити рішення, яке буде достатньо потужним для належного прогнозування цін і гнучким, щоб задовольнити вимоги всіх.

Машинне навчання (ML) - це галузь штучного інтелекту (AI), де комп'ютерні системи навчаються розпізнавати закономірності та приймати рішення на основі даних без явних програмних інструкцій та точно обробляти великі обсяги даних, генеруючи відомості та прогнози з мінімальним втручанням людини [14]. ML дозволяє організаціям раціоналізувати процеси прийняття рішень, підвищити продуктивність та досягти кращих результатів у різних сферах. ML включає багато технік, які дозволяють програмним додаткам покращувати свою продуктивність з часом. Для цього потрібно розуміти математичні та статистичні концепції для вибору відповідних алгоритмів та навчання їх на достатніх даних для досягнення точних результатів.

Машинне навчання для прогнозування цін на житло передбачає використання обчислювальних алгоритмів для аналізу численних факторів, що впливають на вартість нерухомості, таких як характеристики об'єкта, деталі місця розташування, економічні показники та дані про минулі продажі. За допомогою передових статистичних методів та математичних моделей ці алгоритми виявляють закономірності в даних для створення точних прогнозних моделей.

Прикладом використання машинного навчання можна навести дослідження Хлевна Ю. Л. [15], котре вивчає застосування алгоритмів машинного навчання для прогнозування цін на житло шляхом аналізу факторів, таких як місце

розташування, характеристики нерухомості та економічні показники. Автор, збирає та попередньо оброблює великі набори даних про операції з нерухомістю, а потім навчає моделі машинного навчання для прогнозування цін на основі цих факторів. Ключовими проблемами є вибір ознак та вирішення проблеми розрідженості даних, при цьому використовуються такі техніки, як регресія, дерева рішень та агреговані моделі для підвищення точності.

Дослідження “Прогнозування цін на житло з використанням Support Vector Machines”, вивчає використання машин опорних векторів (SVM) для прогнозування цін на житло. Це дослідження охоплює те, як SVM можна навчати на наборах даних про житло для точного прогнозування цін, використовуючи методи попередньої обробки, та налаштування гіперпараметрів, головною метою дослідження є демонстрація ефективності SVM у прогнозуванні нерухомості [16].

Вань. Х. зосереджується на використанні методів аналізу часових рядів для прогнозування майбутніх цін на житло, аналізуючи історичні дані про житло, щоб виявити закономірності та тенденції з часом, застосовуючи такі методи, як моделі ARIMA та експоненційне згладжування. Мета проекту - надати цінну інформацію для розуміння та прогнозування динаміки ринку житла за допомогою аналізу часових даних [17].

Дослідження “Прогнозування майбутніх цін на житло з використанням регресії” вивчає застосування ласо-регресії для прогнозування цін на житло. Мета роботи - продемонструвати ефективність ласо-регресії у прогнозуванні цін на житло та, можливо, надати цінні відомості про оптимальне використання цієї техніки для оцінки вартості нерухомості [18].

В.О. Воронін, М.М. Мамчин, Е.В. Лянце, які є авторами “Прогнозне моделювання тенденцій розвитку ринку нерухомості”, заглиблюється у складнощі прогнозування цін на житло з урахуванням економічних факторів. Через дослідження виявлено, що штучні нейронні мережі є ефективним методом моделювання для прогнозування цін на житло [19].

У прогнозуванні цін на житло лінійна регресія використовується для встановлення прямого зв'язку між залежною змінною (ціною будинку) та набором

незалежних змінних (ознак, таких як кількість спалень, квадратні метри). Алгоритм припускає, що цей зв'язок є лінійним, що означає, що зміна ознаки призводить до пропорційної зміни ціни. Підганяючи лінійне рівняння до історичних даних, модель оцінює коефіцієнти для кожної ознаки, що дозволяє їй прогнозувати ціну будинку, враховуючи його характеристики. Лінійна регресія має кілька обмежень у прогнозуванні цін на житло. По-перше, вона припускає, що зв'язок між ознаками та ціною є лінійним, тоді як насправді він може бути нелінійним. По-друге, вона не може ефективно враховувати взаємодії між ознаками, наприклад, як поєднання високої площі та гарного розташування може збільшити ціну більше, ніж ефект від кожної ознаки окремо. Також лінійна регресія чутлива до викидів у даних та може давати неточні прогнози в присутності аномальних спостережень [20]. Через ці обмеження часто використовуються більш просунуті методи машинного навчання, такі як дерева рішень, випадкові ліси, градієнтний бустинг або нейронні мережі, які можуть краще впоратися зі складними нелінійними залежностями та взаємодіями між ознаками в даних про ціни на житло.

Ridge регресія - це техніка, яка використовується для оцінки наявності колінеарності в даних множинної регресії. Коли виникає мультиколінеарність, оцінки методу найменших квадратів є незміщеними, але їх дисперсії достатньо великі, щоб вони могли значно відхилятися від реального значення. Гребенева регресія забезпечує більш надійну продуктивність за рахунок зменшення стандартних помилок [21]. Ця модель широко використовувалася багатьма дослідниками в різних дослідженнях у сфері нерухомості [22]. Гребенева регресія є корисною для вирішення проблеми мультиколінеарності в даних про ціни на житло, коли незалежні змінні (ознаки) сильно корелюють одна з одною. Це може статися, наприклад, коли площа будинку та кількість спалень сильно пов'язані. Гребенева регресія додає штраф до коефіцієнтів регресії, змушуючи їх бути ближче до нуля, що зменшує вплив мультиколінеарності та робить оцінки більш стабільними. Однак гребенева регресія має свої недоліки. Вона не виконує відбір змінних, тому всі ознаки зберігаються в моделі, що може призвести до перенавчання. Крім того, інтерпретація коефіцієнтів гребеневої регресії може бути

складною через введення штрафу. Тому часто використовуються інші методи регуляризації, такі як лассо-регресія або еластичні мережі, які поєднують переваги гребеневої та лассо-регресії.

XGBoost - це найпотужніший алгоритм для будь-якої задачі регресії або класифікації, що означає "екстремальний градієнтний бустинг". Розробники цієї моделі у своїй статті описують XGBoost як масштабовану систему машинного навчання на основі бустингу дерев. Вплив цієї моделі широко визнаний у різних проблемах машинного навчання та інтелектуального аналізу даних [23].

Згідно з дослідження "Прогнозування ціни на нерухомість з використанням алгоритмів машинного навчання", алгоритм XGBoost вирішує проблеми моделі лінійної регресії [24]. Модель XGBoost добре справляється як з числовими, так і з категоріальними змінними. Ця модель може автоматизувати різні функції втрат і пропонує багато варіантів налаштування для модифікації моделі. Ця модель використовується в багатьох сферах, включаючи ринок нерухомості. Більшість досліджень у галузі нерухомості, які використовують модель XGBoost, виявили, що ця модель здатна забезпечити найнижче значення RMSE для прогнозування цін на житло порівняно з іншими моделями.

Алгоритми дерев рішень особливо корисні для виявлення нелінійних взаємозв'язків та взаємодій між ознаками при прогнозуванні цін на житло. Модель розділяє дані на гілки на основі рішень, пов'язаних з різними ознаками, такими як місце розташування або тип нерухомості. У кожному вузлі алгоритм вибирає ознаку, яка найкраще розділяє дані на підмножини, де ціни на житло відносно однорідні. Цей процес продовжується, поки дерево не досягне максимальної глибини або подальші розділення не покращать точність моделі [25]. Результуюча деревоподібна структура забезпечує інтуїтивно зрозумілу та інтерпретовану модель, яка прогнозує ціни на житло, слідуючи шляхом прийняття рішень від кореня до листових вузлів. Однак дерева рішень можуть перенавчатися на даних, особливо при роботі зі складними наборами даних, що призводить до менш надійних прогнозів на нових даних.

Випадковий ліс покращує дерева рішень, створюючи ансамбль кількох дерев рішень, кожне з яких навчається на різних підмножинах даних. При прогнозуванні цін на житло модель випадкового лісу агрегує прогнози всіх окремих дерев, як правило, усереднюючи їх, для отримання остаточної оцінки. Такий підхід зменшує дисперсію та знижує ризик перенавчання, від якого можуть страждати окремі дерева рішень. Алгоритм випадкового лісу дуже ефективний в роботі з великими та складними наборами даних, що робить їх придатними для моделювання різноманітних факторів, які впливають на ціни на житло. Крім того, випадковий ліс надає інформацію про важливість ознак, виділяючи фактори, які мають найбільший вплив на ціну [26].

Основна перевага алгоритму випадкового лісу полягає в тому, що цей алгоритм поєднує прогнози кількох незалежних моделей, зменшуючи тим самим дисперсію та ризик перенавчання. Кожне окреме дерево рішень має свою унікальну упередженість, але шляхом усереднення прогнозів цих різноманітних дерев випадковий ліс може компенсувати індивідуальні упередженості. Крім того, випадковий ліс вводить додаткову випадковість у процес побудови дерев, що допомагає зменшити перенавчання. Для кожного дерева випадково вибирається підмножина ознак, на основі яких будується дерево. Це призводить до створення різноманітних дерев, які роблять різні припущення про важливість ознак.

Використовуючи ці моделі, зацікавлені сторони в секторі нерухомості можуть приймати обґрунтовані рішення щодо інвестицій у нерухомість, підходів до продажу та ринкових тенденцій, що призводить до підвищення ефективності та оптимізації результатів на ринку житла. Загалом, впровадження технік машинного навчання в системи прогнозування цін на житло є значним проривом у цій галузі, пропонуючи підвищену точність, ефективність та адаптивність для зацікавлених сторін на ринку нерухомості.

### **1.3 Теоретичні основи Data Driven підходу та його роль у прийнятті рішень в економіці**

У сучасному світі, де дані стають все більш доступними, прийняття рішень на основі даних набуває критичного значення для успіху як організацій так і бізнесу. Рушійною силою сучасної науки про дані є наявність величезної та зростаючої кількості даних, забезпечена надзвичайними інноваціями, збільшенням обчислювальної потужності та практично необмеженими можливостями зберігання та передачі даних. Такі величезні обсяги даних дають інженерам та вченим у всіх дисциплінах нові можливості для відкриттів, заснованих на даних, що називається четвертою парадигмою наукових відкриттів. Ця четверта парадигма є природним завершенням перших трьох парадигм: емпіричного експериментування, аналітичного виведення та обчислювального дослідження. Інтеграція цих технік забезпечує трансформаційну основу для зусиль з відкриттів, заснованих на даних. Цей процес наукового відкриття не є новим і справді імітує зусилля провідних фігур наукової революції: Йоганнеса Кеплера (1571-1630) та сера Ісаака Ньютона (1642-1727). Кожен з них відіграв критичну роль у розробці теоретичних основ механіки на основі поєднання емпіричних та аналітичних підходів, заснованих на даних. Наука про дані не замінює математичну фізику та інженерію, а натомість доповнює її для XXI століття [27]. Сама наука про дані не є новою, оскільки була запропонована понад 50 років тому Джоном Тьюкі, який передбачив існування наукових зусиль, зосереджених на навчанні з даних, або аналізі даних. З того часу наука про дані значною мірою домінувалася двома різними культурними поглядами на дані [28].

Прийняття рішень на основі даних (DDDM) – це процес ухвалення бізнес-рішень, що ґрунтується на інсайтах, отриманих з аналізу даних. Цей підхід виявився корисним для компаній будь-якого розміру, бізнес-моделі чи галузі, оскільки він дозволяє приймати більш обґрунтовані та точні рішення швидко й ефективно. DDDM особливо ефективний у сферах маркетингу, продажів, обслуговування клієнтів та управління ланцюгами поставок, де він допомагає

виявляти тенденції, аналізувати поведінку споживачів, вимірювати продуктивність і робити точніші прогнози. Він також застосовується в багатьох інших бізнес-напрямах, таких як фінанси, управління персоналом і розробка продуктів [29].

Прийняття рішень на основі даних допомагає знизити ризики для бізнесу. Завдяки використанню достовірних даних компанії можуть приймати більш виважені та точні рішення, на відміну від ситуацій, коли рішення ухвалюються без належної інформаційної бази. Відстежуючи ключові показники ефективності та метрики, організації можуть швидко ідентифікувати потенційні проблемні зони та вжити необхідних заходів для їх виправлення. Це дозволяє мінімізувати ймовірність помилок або повністю їх уникнути.

Крім того, DDDM може допомогти скоротити витрати, визначивши сфери неефективності, що дозволить компаніям оптимізувати свою діяльність і підвищити прибутковість. Це особливо актуально в таких сферах, як управління запасами та оптимізація ціноутворення, де рішення на основі даних можуть допомогти бізнесу зменшити витрати, пов'язані з надмірним запасом або заниженою ціною на продукцію.

Одна з переваг прийняття рішень на основі даних - це прискорення самого процесу ухвалення рішень. Завдяки використанню фактичних даних замість інтуїції, компанії можуть приймати рішення швидше. Це дозволяє їм оперативно реагувати на мінливі ринкові умови та зберігати конкурентоспроможність. Здатність приймати своєчасні рішення особливо цінна в ситуаціях, коли фактор часу є критичним, наприклад, при необхідності негайно відреагувати на скаргу клієнта або вирішити кризову ситуацію.

Прийняття рішень на основі даних дозволяє компаніям робити більш точні прогнози щодо майбутніх результатів, розуміючи поточні тенденції та аналізуючи дані про минулі результати діяльності. Це може допомогти бізнесу планувати майбутнє зростання та підготуватися до будь-яких непередбачуваних подій у поінформований спосіб.

Завдяки інсайтам на основі даних компанії можуть точніше вимірювати рентабельність інвестицій своїх маркетингових кампаній і вносити дієві зміни в

маркетингові стратегії за потреби. Це допомагає їм максимізувати маркетингові витрати та забезпечити їх ефективне використання. З точки зору маркетингу, DDDM також дозволяє компаніям визначити, які маркетингові канали є найбільш ефективними, і відповідно скоригувати свої маркетингові стратегії. Це допомагає їм охопити потрібну аудиторію з потрібним повідомленням – і в потрібний час, що призводить до збільшення рентабельності інвестицій в маркетинг.

Для успішного впровадження системи прийняття рішень на основі даних компаніям необхідний доступ до надійних і точних даних. Ці дані повинні бути легкодоступними та зрозумілими для всіх зацікавлених сторін, щоб вони могли приймати обґрунтовані рішення. Інструменти візуалізації даних можуть допомогти перетворити дані на корисні інсайти (рис.1.1).



Рисунок 1.1 – Етапи Data-Driven підходу

*Джерело: розроблено автором на основі [30]*

Визначення потреб у даних є ключовим етапом у Data Driven підході. Одним із найсерйозніших порушень при прийнятті рішень на основі даних є збирання даних без чіткої мети та плану їх використання. Щоб уникнути такої помилки, завжди слід починати планування прийняття рішень на основі даних з визначення потреб у даних. Потреба в даних - це те, що організація намагається досягти або зробити, для чого потрібні дані. Понад усе, визначення потреб у даних стосується

можливості отримати розуміння, щоб потім вжити відповідних дій. Один із способів визначити потребу в даних - відповісти на запитання: "Чому потрібні ці дані?". Після визначення потреби у даних, наступним кроком є визначення джерела даних: які дані відповідатимуть на запит.

Data-Savvy є компонентом Data-Driven підходу, який базується на використанні емпіричних даних для прийняття рішень та оптимізації бізнес-процесів. На технічному рівні Data-Savvy передбачає – здатність читати, інтерпретувати та критично оцінювати структуровані та неструктуровані дані з різних джерел. Це включає розуміння метаданих, можливих упереджень та обмежень даних [31].

Кількісні дані – це дані у вигляді чисел, для того аби зрозуміти кількісні дані та мати змогу використовувати їх, потрібні великі обсяги даних котрі повинні бути зібрані протягом тривалого періоду часу. Багато в чому аналізувати кількісні дані простіше, ніж якісні дані, оскільки вони не вимагають подальшої категоризації чи індивідуального вивчення записів даних. Перевага кількісних даних полягає в тому, що візуалізувати дані на графіках набагато простіше, це певним чином полегшує розуміння того, що відбувається. Також зазвичай легше накладати та перехресно аналізувати кількісні дані, що може дійсно допомогти зробити більш цінні висновки з ваших даних [32].

Якісні дані - це тип даних, який представлений не числами, а словами, категоріями та описами. На відміну від кількісних даних, які є числовими, якісні дані є описовими та не кількісними за своєю природою. Якісні дані можуть надати додаткову цінну інформацію. Наприклад, якісні дані можуть включати описи стану житлового комплексу, опис об'єкта нерухомості, наявність інфраструктури поблизу, такі як школи, парки чи торгові центри. Ці описові дані можуть допомогти краще зрозуміти загальну привабливість нерухомості та фактори, які можуть вплинути на її ціну. Крім того, якісні дані можуть містити відгуки покупців або експертні оцінки професіоналів у галузі нерухомості. Ці думки та судження можуть надати цінну інформацію про сприйняття певної нерухомості на ринку, що може вплинути на її ціну. Наприклад, якщо в певному районі є високий попит через престижність або

зручне розташування, це може відобразитися у відгуках покупців та експертів, що може бути корисним для прогнозування цін.

Якісні дані також можуть включати інформацію про плани розвитку в районі, де розташована нерухомість. Наприклад, якщо планується будівництво нової транспортної розв'язки або торгового центру поблизу, це може вплинути на привабливість та ціни на нерухомість у цьому районі.

Аналізуючи та інтегруючи якісні дані з кількісними даними, такими як площа, кількість кімнат та місце розташування, можна створити більш повну та точну модель прогнозування цін на нерухомість. Кодування та категоризація якісних даних, а також виявлення можуть допомогти краще зрозуміти фактори, що впливають на ціни на нерухомість, та покращити точність прогнозів.

Аналіз - це дія обробки даних, виділення та витягування відповідних даних для висновку. Обробка даних може охоплювати безліч завдань та змін у даних, які потрібно виконати, щоб мати змогу робити висновки. Етап очищення та перетворення даних, використовується щоб усунути помилки, пропущені значення та привести дані до структурованого формату, придатного для аналізу. Цей етап включає в себе видалення дублікатів, нормалізацію даних та перетворення форматів. Окрім обробки даних та всіх процесів, пов'язаних із підготовкою даних до аналізу, також потрібно розглянути дані, визначити будь-які аномалії та спробувати їх зрозуміти.

Аномалії - це одноразові та екстремальні піки або провали у даних, які не відповідають регулярному шаблону решти даних. По суті, це дані, які явно відхиляються від стандарту, встановленого рештою даних. Щоб виявити будь-які аномалії, потрібно побудувати графіки даних та визначити все, що здається незвичайним.

Після того, як було виявлено аномалії, наступним кроком є визначення причини виникнення аномалій або того, що їх створило. Часто в даних є помилки, які потрібно виправити, наприклад, неправильно введені або відсутні дані. Якщо аномалія не є помилкою, потрібно визначте, які інші фактори призвели до цього

піку або падіння, потім вирішити, що насправді потрібно робити з нею на основі розуміння аномалії [33].

Візуалізація результатів та формулювання висновків є необхідними етапами в процесі аналізу даних. Після ретельного очищення, обробки та аналізу даних, необхідно ефективно представити отримані результати та зробити змістовні висновки, які можуть бути використані для прийняття обґрунтованих рішень або розробки стратегій. Візуалізація даних допомагає у полегшенні інтерпретації складних наборів даних та виявленні прихованих закономірностей. Ефективні візуалізації перетворюють сирі числа та статистику на зрозумілі зображення, такі як графіки, діаграми. Вони дозволяють швидко виявляти тенденції або взаємозв'язки, які можуть бути непомітними в табличних даних [34].

Після створення візуалізацій наступним кроком є формулювання висновків на основі отриманих результатів. Цей процес вимагає аналізу візуалізацій, виявлення ключових закономірностей та тенденцій, а також інтерпретації їх значення та наслідків. Висновки повинні бути чіткими, зрозумілими та підкріплюватися даними, представленими у візуалізаціях.

Після реалізації аналізу даних та висновків, наступним кроком є дія, крок дії - це впровадження того, що необхідно зробити для реалізації речей, визначених у висновках. Аналіз даних визначає, що відбувається, висновок - вирішення, що потрібно зробити з цим, а дія - це фактично подальша реалізація висновків.

Виконання дій - це свідоме прийняття рішення та плану щодо того, що робити з інформацією, яка була отримана, та змін, необхідних для вдосконалення певної цільової задачі. Частиною кроку дії в процесі прогнозування ціни на об'єкт нерухомості є реалізація планів на основі всієї роботи, проведеної до цього моменту. Приділення часу для належного планування - це найкращий спосіб переконатися, що було обрано найкращий підхід до впровадження дій.

Підхід до впровадження дій як до проекту, може допомогти забезпечити ефективність дій та сприяти кращій роботі та швидкій інтеграції в практику прогнозування цін на нерухомість. Навіть найменші дії, вжиті на основі підходу прийняття рішень на основі даних, потребують точного планування. Спочатку

потрібно розбити дію на кроки, які необхідно виконати для реалізації. Прикладом розбиття дії на складові може бути використання певної архітектури нейронної мережі для прогнозування ціни на об'єкт нерухомості. Це може вимагати допоміжних дій, наприклад:

1. Вибір відповідної архітектури нейронної мережі
2. Визначення гіперпараметрів моделі
3. Тренування моделі на наборі даних
4. Оцінка продуктивності моделі на тестовому наборі даних
5. Налаштування та оптимізація моделі для покращення результатів
6. Розгортання оптимізованої моделі для прогнозування цін на практиці

Також, може знадобитися додаткове планування для управління однією чи кількома з цих дій, де виникатимуть подальші завдання. Необхідно пропрацювати їх усі, поки не буде набір чітких і окремих дій для реалізації. Після визначення окремих дій їх слід розставити за пріоритетами - високий, середній та низький - відповідно до їх важливості для успішного прогнозування цін за допомогою нейронних мереж. Хоча всі дії потрібно виконати для досягнення кінцевої мети, розставлення пріоритетів допоможе організувати та зосередитися на тому, які ресурси та час потрібно виділити для кожної з них, а також на порядку їх виконання [35].

Грунтовне розуміння теоретичних засад Data Driven підходу, його витоків, фундаментальних принципів та ключових етапів реалізації є критично важливим для усвідомлення справжньої значущості та величезного потенціалу цього підходу.

Слід розуміти, що Data Driven підхід не є ізольованою концепцією – він тісно взаємодіє з іншими галузями знань, науковими дисциплінами та методологіями. Зокрема, Data Driven підхід має зв'язок з теорією прийняття рішень – сферою, яка вивчає процеси, методи та інструменти для вибору найкращих альтернатив у складних ситуаціях з урахуванням множини факторів, критеріїв та обмежень.

Прийняття рішень є сучасною прикладною науковою дисципліною, яка досліджує теорію та методи вирішення складних задач вибору. Складність та багатоаспектність цього процесу зумовлює міждисциплінарний характер

досліджень у галузі прийняття рішень, де використовуються підходи та результати математики, штучного інтелекту, інформатики [36]. Поняття "рішення" може трактуватися по-різному: як процес і як результат вибору. Рішення як процес являє собою акт вибору, що здійснюється особою, яка приймає рішення, за допомогою певних правил. Рішення як результат вибору являє собою певний припис до дії.

Рішення характеризується такими ознаками:

- можливість вибору з альтернативних варіантів
- наявність мети: безцільний вибір не розглядається як рішення;

За виконуваними завданнями та умовами вироблення рішення поділяються на три групи:

- ті, що забезпечують функціонування діючої системи;
- спрямовані на оперативні зміни в рамках існуючої системи;
- спрямовані на подальший розвиток та зміну системи.

Кількість розглянутих альтернатив також є важливим критерієм класифікації рішень. Вони можуть бути бінарними (лише дві альтернативи), стандартними (обмежений набір альтернатив), багатоальтернативними (велика, але скінченна кількість варіантів) або безперервними (вибір з нескінченної множини змінних величин) [37].

Використання теорії та методів прийняття рішень дуже ефективно при вирішенні слабо структурованих задач. Слабо структурованими задачами називають такі, в яких багато змінних є змінними якісного порядку. Задача є добре структурованою, якщо для неї визначені (на множині дійсних чисел) значення всіх змінних, погано структурованою - якщо в ній практично всі змінні - якісного порядку.

Процес прийняття рішень є надзвичайно складним не лише за змістом його етапів, але й за змістом прямих та непрямих взаємозв'язків цих етапів. Виявлення та представлення цих взаємозв'язків є основним завданням складання моделей процесу прийняття рішень. Звичайно, кожна модель істотно спрощує процес, який моделюється.

У процесі прийняття рішень можна виділити такі елементи:

- проблемна ситуація
- час для прийняття рішень
- необхідні для прийняття рішення ресурси
- система (вектор) цілей, що переслідуються при прийнятті рішень
- система керованих факторів, з комбінування станів яких формуються альтернативні варіанти рішень

У теорії прийняття рішень ціль та альтернативи зазвичай розглядаються як задані ззовні величини. Ціль визначають як кінцевий стан зміни керованого процесу. Одна й та сама дія може розглядатися і як ціль, і як альтернатива залежно від рівня задачі. Так, у дереві цілей підцілі першого рівня є альтернативами досягнення головної мети, але для другого рівня вони є цілями [38].

Дерево взаємозв'язків може являти собою повний зв'язний граф (містити цілі, ресурси) або бути незв'язним графом (містити або цілі, або ресурси). У першому випадку дерево містить рівні цілей, рівні заходів та рівні ресурсів. При цьому заходи слід конкретизувати для кожної конкретної мети, представленої на останньому рівні дерева цілей.

Виділяють різні етапи вирішення процесу прийняття рішень, також варто звернути увагу, що в загальному випадку процес може мати ітераційний характер з неодноразовими поверненнями з будь-якого етапу на будь-який попередній. На етапі виявлення та аналізу проблемної ситуації аналізується інформація про стан об'єкта дослідження та зовнішнього середовища, виявляються базові проблеми з їх ранжуванням за значущістю, визначаються причини виникнення проблем та стратегічні напрямки вирішення проблем. Даний етап повинен давати відповіді на такі питання: в чому полягає проблема? чи потрібно її вирішувати? в яких умовах потрібно її вирішувати? якими силами і засобами? які необхідні ресурси для вирішення проблеми?

Формулювання цілей прийняття рішень. Мета являє собою складну систему уявлень про бажаний результат діяльності. Мета повинна бути конкретною та зрозумілою, аби сфокусуватися на бажаному результаті.

Наступним кроком є формування системи критеріїв для оцінки альтернативних варіантів рішень. Потрібно виділити найбільш важливі показники, які характеризують кожну альтернативу, щоб мати змогу порівняти їх між собою.

Перед тим, як приступити до безпосереднього вирішення завдання, необхідно провести підготовку. Це включає вибір методу кількісної оцінки критеріїв та визначення шкал вимірювання їхніх значень. Також важливо встановити допустимі межі зміни значень критеріїв.

Формування множини альтернативних варіантів рішень. Цей етап передбачає отримання числових значень критеріїв для всіх альтернатив різними способами. Чим більша множина альтернатив, тим більше шансів, що найкраще в даних конкретних умовах рішення буде знайдено.

Наступним етапом йде визначення методів рішення. Виконується вибір або розробка, по-перше, допустимих і, по-друге, найбільш ефективних для даного завдання методів пошуку рішення.

Під час прийняття рішень важливо ретельно розглянути всі можливі альтернативи. Перш за все, необхідно провести попередню обробку альтернатив, щоб відсіяти ті варіанти, які не відповідають встановленим обмеженням або формальним вимогам. Цей крок допомагає звужити коло пошуку та зосередитися на найбільш релевантних альтернативах.

Після попереднього відсіювання альтернатив настає етап вирішення завдання. На цьому етапі здійснюється власне аналіз альтернатив та подальше звуження їх множини. Це центральний момент у процесі прийняття рішень, і його можна виконувати за допомогою нейронних мереж для задачі прогнозування. Під час аналізу альтернатив можна одразу зробити вибір єдиної найкращої альтернативи або впорядкувати альтернативи за перевагами [39].

У цьому контексті Data Driven підхід постає як потужна методологія прийняття рішень, яка ґрунтується на аналізі та інтерпретації емпіричних даних, а не на інтуїції чи суб'єктивних думках. Це підхід, який розглядає дані як найцінніший актив організації та передбачає їх ефективне використання для отримання конкурентної переваги та покращення процесів прийняття рішень.

Замість покладання на припущення чи досвід, організації, які використовують Data Driven підхід, покладаються на емпіричні дані для інформування своїх стратегій та рішень. Цей підхід передбачає аналіз великих обсягів даних з різних джерел для виявлення закономірностей, тенденцій та взаємозв'язків. Результати аналізу даних інтерпретуються для прийняття обґрунтованих рішень, які можуть стосуватися вдосконалення, оптимізації процесів, розробки нових стратегій або інших аспектів [40].

Data Driven підхід є ітеративним процесом, який передбачає постійний моніторинг та вдосконалення. Потрібно регулярно збирати нові дані, аналізувати їх та коригувати свої рішення відповідно до нових знань. Цей підхід має ряд переваг, включаючи об'єктивність, ефективність, можливість прогнозування та прийняття більш обґрунтованих рішень.

Використання підходу прийняття рішень на основі даних може допомогти досягти кількох ключових переваг. По-перше, воно дозволяє ретельно проаналізувати та оцінити ефективність та результативність наданих послуг. Замість покладання на суб'єктивні думки чи застарілі припущення, організації можуть використовувати фактичні дані для об'єктивного вимірювання успіху своїх зусиль та визначення областей, які потребують вдосконалення. Крім того, прийняття рішень на основі даних дозволяє виявляти зони для покращення або успішні практики, які можна відтворити в інших сферах діяльності. Шляхом аналізу даних організації можуть ідентифікувати чинники, що сприяють успіху певних ініціатив або продуктів, та застосувати ці уроки для підвищення ефективності в інших областях.

Ще одним важливим аспектом є здатність виявляти зміни потреб та тенденцій на ринку. Регулярний моніторинг та аналіз відповідних даних дозволяє швидко реагувати на мінливі вподобання, технологічні зрушення або інші зміни в середовищі. Це дає змогу розробляти нові послуги, продукти або стратегії, які відповідають поточним та майбутнім потребам.

Отже, оскільки метою дипломної роботи є обґрунтування вибору об'єктів нерухомості з використанням підходу, заснованого на даних, важливо розглянути

різні методи та інструменти, які можна застосувати для вирішення цього завдання. Одним з перспективних підходів є використання нейронних мереж - потужних моделей машинного навчання, здатних виявляти складні закономірності в даних.

У контексті теорії прийняття рішень вибір нейронних мереж як засобу для прогнозування цін на нерухомість можна розглядати як результат процесу прийняття рішень, в якому враховуються різні критерії та альтернативи. Альтернативами можуть бути різні архітектури нейронних мереж, різні типи оптимізації, та різні функції активації нейронних мереж. Критеріями для оцінки цих альтернатив можуть бути точність прогнозування та швидкість навчання моделі.

Таким чином, розгляд нейронних мереж як засобу для прогнозування цін на об'єкти нерухомості є результатом процесу прийняття рішень, заснованого на теорії прийняття рішень, в якому враховуються різні критерії та альтернативи. Це дозволяє обґрунтувати вибір нейронних мереж як ефективного інструменту для вирішення поставленого завдання та забезпечити систематичний підхід до розробки та впровадження моделі прогнозування цін.

## РОЗДІЛ 2

### АНАЛІЗ МЕТОДІВ АВТОМАТИЗОВАНОГО МАШИННОГО ТА ГЛИБОКОГО НАВЧАННЯ ДЛЯ ОЦІНЮВАННЯ ВАРТОСТІ ОБ'ЄКТІВ ПРИВАТНОГО ЖИТЛОВОГО ІНВЕСТИВАННЯ

#### 2.1 Інструментарій обґрунтування рішень при виборі приватних житлових об'єктів на базі аналітики даних та машинно-навчальних технологій моделювання

Машинне навчання — це галузь штучного інтелекту, яка зосереджується на розробці алгоритмів та моделей, здатних автоматично вчитися та покращувати свої можливості на основі даних. Машинне навчання відіграє ключову роль у сучасних технологіях та має широке коло застосувань у різних галузях, таких як комп'ютерний зір, обробка природної мови, рекомендаційні системи, фінансовий аналіз та багато інших [41].

Машинне навчання поділяється на три основні типи: навчання з вчителем, навчання без вчителя та навчання з підкріпленням, кожен з яких має свої унікальні переваги та застосування (рис. 2.1).

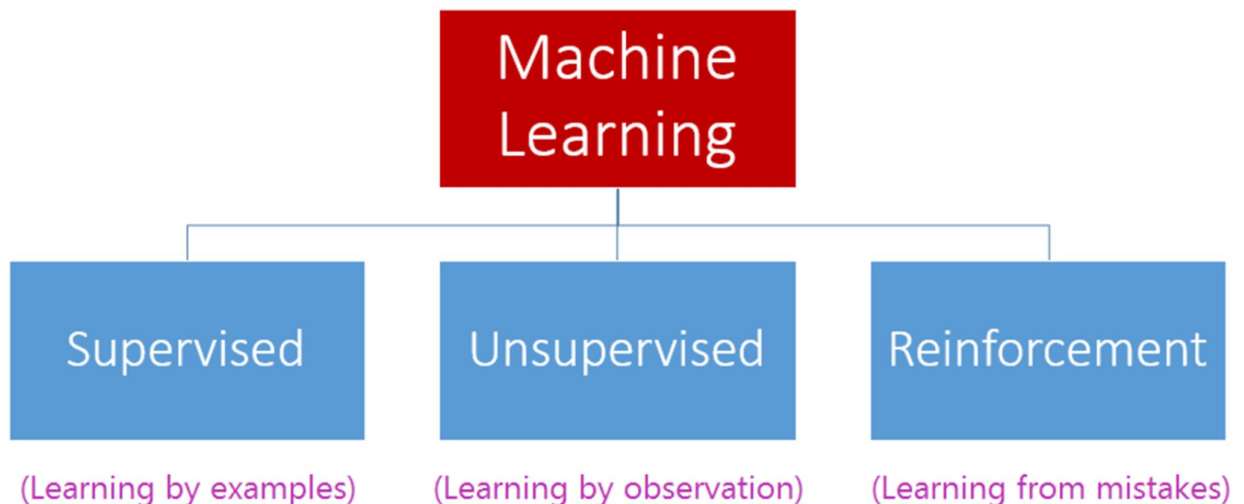


Рисунок 2.1 – Типи навчання в Machine Learning

*Джерело: розроблено автором на основі [42]*

Навчання з вчителем (Supervised Learning) є найбільш поширеним і добре зрозумілим типом машинного навчання. У цьому підході алгоритм навчається на розміченому наборі даних, де вхідні дані (ознаки) пов'язані з відповідними виходами (мітками). Наприклад, алгоритм може навчитися передбачати ціни на житло, використовуючи ознаки, такі як розмір, місце розташування та кількість кімнат, а мітки - це фактичні ціни на нерухомість. Після навчання на великій кількості прикладів модель може робити точні прогнози для нових, невідомих даних. Перевагою навчання з вчителем є його здатність вирішувати складні завдання класифікації та регресії з високою точністю, що робить його ідеальним для таких застосувань, як розпізнавання зображень, обробка природної мови та прогнозування продажів. Загалом, навчання з учителем застосовується в задачах, коли необхідно отримати результат для наданих даних, спираючись на минулий досвід. Прикладом задачі навчання з вчителем є класифікація, класифікація - це група методів мета яких - присвоїти дискретну мітку класу з наперед визначеного обмеженого набору варіантів (двох або більше). Наприклад, система для моніторингу фінансових транзакцій та виявлення шахрайства. Для навчання такої системи потрібні історичні дані, що містять як звичайні, так і шахрайські транзакції, з чітким позначенням, які рядки даних є шахрайськими. Інший поширений приклад - система аналізу тональності тексту, яка класифікує вхідний текст як "позитивний", "негативний" або "нейтральний". Регресія - метод навчання з вчителем, який намагається встановити зв'язок між двома змінними. У задачах регресії передбачається безперервне дійсне значення, тобто цільова змінна може приймати нескінченну кількість значень, на відміну від класифікації, де присутня обмежена кількість фіксованих класів. Прикладом регресії є система для прогнозування вартості акцій на наступний день, використовуючи дані про ціну та обсяг торгів попереднього дня.

З іншого боку, навчання без вчителя (Unsupervised Learning) не вимагає розмічених даних. Замість цього алгоритм досліджує дані, щоб виявити приховані закономірності та структури. На відміну від навчання з вчителем де алгоритми навчаються на основі розмічених даних, навчання без вчителя дозволяє моделям

виявляти приховані закономірності та структури в неструктурованих даних без попереднього розмічення. Основна мета навчання без вчителя полягає в тому, щоб знайти корисні закономірності та взаємозв'язки в даних, які можуть бути використані для різних завдань, таких як кластеризація, зменшення розмірності, виявлення аномалій та генерація даних.

Одним з найбільш поширених завдань навчання без вчителя є кластеризація, яка групує подібні точки даних у кластери. Алгоритми кластеризації, такі як k-means, ієрархічна кластеризація, широко використовуються в різних галузях, таких як сегментація клієнтів та аналіз даних, кластеризація допомагає виявляти природні групи або кластери в даних без попереднього знання про їхню структуру.

Навчання без вчителя також використовується для зменшення розмірності даних, що полегшує візуалізацію та обробку великих наборів даних. Перевагою цього підходу є його здатність виявляти нові закономірності та отримувати цінні знання з необроблених даних без необхідності розмічати їх вручну. Виявлення аномалій є ще одним напрямком навчання без вчителя. Алгоритми виявлення аномалій навчаються розпізнавати незвичайні або відхилені зразки в даних, що має величезне значення для моніторингу мережі та діагностики несправностей.

Нарешті, навчання з підкріпленням (Reinforcement Learning) є унікальним типом машинного навчання, де алгоритм вчиться шляхом взаємодії з середовищем. Замість того, щоб використовувати розмічені дані, агент (модель) виконує дії в середовищі та отримує винагороди або штрафи залежно від успішності своїх дій. Мета полягає в тому, щоб максимізувати загальну винагороду, змушуючи агента вдосконалювати свою стратегію з часом. Цей підхід широко використовується в керуванні роботами та системах рекомендацій. Перевагою навчання з підкріпленням є його здатність вирішувати складні завдання прийняття рішень у динамічних середовищах, де правила не завжди чітко визначені (рис. 2.2) [43].

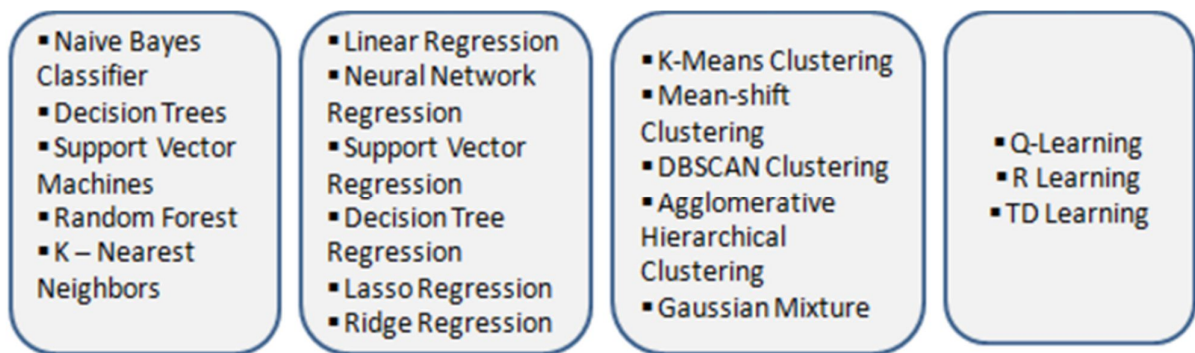


Рисунок 2.2 – Базові алгоритми Machine Learning

*Джерело: розроблено автором на основі [44]*

Однією з найважливіших концепцій машинного навчання є розуміння ознак та міток. Ознаки - це вхідні змінні, які подаються в модель машинного навчання. Вони можуть бути числовими, категоріальними або навіть неструктурованими даними, такими як текст або зображення. Мітки, з іншого боку, є бажаним результатом або виходом, який модель повинна передбачити або класифікувати на основі вхідних ознак. Під час навчання модель вчиться відображати ознаки на відповідні мітки, встановлюючи зв'язки та закономірності в даних.

Наступною концепцією є процес навчання та тестування. Для навчання моделі використовується підмножина наявних даних, яка називається навчальним набором. Модель аналізує ці дані та налаштовує свої внутрішні параметри, щоб якнайкраще відображати ознаки на мітки. Однак, щоб оцінити реальну продуктивність моделі, її необхідно протестувати на невідомих даних, які не використовувалися під час навчання. Ця підмножина даних називається тестовим набором. Порівнюючи передбачення моделі з фактичними мітками в тестовому наборі, можна визначити її точність та ефективність [45].

Однією з найбільших проблем у машинному навчанні є перенавчання та недонавчання. Перенавчання відбувається, коли модель стає надто складною та близько підігнаною до навчальних даних. У такому випадку вона "запам'ятовує" специфічні деталі та шум у навчальних даних, що призводить до поганої продуктивності на нових, невідомих даних. З іншого боку, недонавчання трапляється, коли модель є занадто простою і не здатна виявити основні закономірності в даних. Це призводить до того, що модель не може адекватно

узагальнити та робити точні передбачення. Для вирішення цих проблем використовуються різні техніки регуляризації, такі як зменшення розмірності даних, застосування штрафів до складності моделі та використання методів ансамблювання. Ці методи допомагають знайти баланс між здатністю моделі узагальнювати та її здатністю підігнати дані, забезпечуючи оптимальну продуктивність [46].

Машинне навчання для прогнозування цін на житло передбачає використання обчислювальних алгоритмів для аналізу численних факторів, що впливають на вартість нерухомості, таких як характеристики об'єкта, деталі місця розташування, економічні показники та дані про минулі продажі. За допомогою передових статистичних методів та математичних моделей ці алгоритми виявляють закономірності в даних для створення точних прогнозних моделей.

## **2.2 Використання нейромережевого моделювання для оцінювання вартості житла**

Штучна нейронна мережа – це математична модель, побудована на основі функціонування нейронів головного мозку, яка застосовується для вирішення різноманітних прикладних завдань. Вона імітує роботу нейронів головного мозку - простих однотипних обчислювальних елементів, які, обмінюючись сигналами між собою, здатні виконувати складні завдання.

Історія розвитку нейронних мереж бере свій початок ще у 1940-х роках, коли вчені почали досліджувати можливість створення штучних систем, що наслідують роботу біологічного мозку. Ця ідея була натхненна розумінням того, що мозок людини є надзвичайно потужною та ефективною обчислювальною системою, здатною виконувати складні завдання, такі як розпізнавання образів, навчання та прийняття рішень [47].

Штучні нейронні мережі існують досить довго: вперше вони були введені в далекому 1943 році нейрофізіологом Уорреном Мак-Каллоком та математиком

Уолтером Піттсом. Мак-Каллок і Піттс представили спрощену обчислювальну модель того, як біологічні нейрони можуть працювати разом для виконання складних обчислень із застосуванням логіки висловлювань (propositional logic). Це була перша архітектура штучної нейронної мережі.

У 1958 році Френк Розенблатт розробив одну з перших успішних нейронних мереж, яку він назвав "персептрон". Персептрон був здатний навчатися розпізнавати прості геометричні фігури та символи, що продемонструвало потенціал нейронних мереж для вирішення задач розпізнавання образів [48].

Перед початком ознайомлення з штучним нейроном, та штучною нейронною мережею, слід ознайомитися з структурою біологічного нейрона.

Структура біологічного нейрона має наступний вид: Через дендрити (dendrites) в ядро надходять електричні сигнали (рис. 2.3), де починає накопичуватися електричний заряд. Коли він досягає певного рівня, нейрон збуджується (fire) і посиляє електричний сигнал через аксон (аксон).

Аксон є довгим відростком нейрона, який передає електричні імпульси до інших нейронів або м'язових клітин. На кінці аксона знаходяться синапси - спеціалізовані структури, які дозволяють передавати сигнал до інших клітин. Коли електричний імпульс досягає синапсів, він викликає вивільнення нейротрансмітерів - хімічних речовин, які переносять сигнал через синаптичну щілину до наступного нейрона або м'язової клітини [49].



Рисунок 2.3 – Будова нейрона

*Джерело: розроблено автором на основі [49]*

Таким чином, нейрони утворюють складну мережу, в якій електричні та хімічні сигнали передаються від одного нейрона до іншого, забезпечуючи обробку

інформації та координацію різних функцій організму. Ця мережа лежить в основі всіх процесів, пов'язаних з сприйняттям, рухом, пам'яттю, емоціями та когнітивними функціями.

Однією з ключових відмінностей між біологічними та штучними нейронними мережами є швидкість передачі сигналів та навчання. Біологічні нейронні мережі здатні передавати імпульси з високою швидкістю, яка може сягати до 120 метрів за секунду. Натомість, штучні мережі обмінюються інформацією зі швидкістю, що обмежена продуктивністю сучасних комп'ютерних процесорів. Крім того, штучні нейромережі можуть навчатися значно швидше, ніж біологічні, завдяки можливості використовувати паралельні алгоритми та оптимізацію обчислень. У той же час біологічні нейронні мережі мають набагато більшу кількість нейронів і зв'язків, що дає їм перевагу в аналізі складних ситуацій і формуванні адаптивних стратегій. Незважаючи на досягнення, штучні нейронні мережі не можуть повністю відтворити всі функції людського мозку.

Зрозуміти принцип роботи штучної нейронної мережі можна на прикладі персептрона - однієї з перших моделей нейронної мережі. Модель персептрона представлена на рис. 2.4.

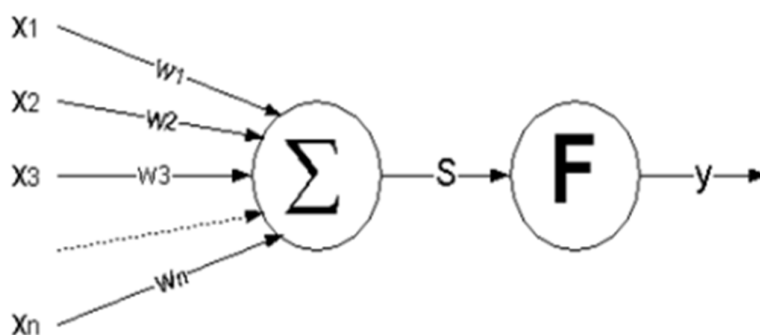


Рисунок 2.4 – Структура персептрона

*Джерело: розроблено автором на основі [50]*

Персептрон (Perceptron) - одна з найпростіших архітектур штучної нейронної мережі, придумана Френком Розенблаттом у 1957 році. В експерименті Розенблатта

вхідні дані та вихід були числами, ознаками слугували значення яскравості кожного з пікселів (фрагментів зображення). Далі, для кожного вхідного значення, присвоювалася асоційована вага [див. Формулу (2.1)] [50].

$$y = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 \dots x_n \times w_n \quad (2.1)$$

*де  $x$  – вхідний сигнал;*

*$w$  - значення ваг кожного нейрону;*

*$y$  – вихідний сигнал.*

Персептрон обчислює зважену суму вхідних сигналів та застосовує до отриманої суми функцію активації (рис 2.5). Найпоширенішою ступінчастою функцією, що використовується в персептронах, є ступінчаста функція Хевісайда (Heaviside step function).

$$\text{heaviside}(z) = \begin{cases} 0, & \text{если } z < 0 \\ 1, & \text{если } z \geq 0 \end{cases}$$

Рисунок 2.5 – Функція активації Хевісайда

*Джерело: розроблено автором на основі [50]*

Таким чином, персептрон видає 1, якщо зважена сума входів додатня, і 0 у протилежному випадку. Це найпростіша форма бінарного класифікатора.

Математична функція, яка визначає вихідний сигнал нейрона, називається активаційною функцією. Активаційна функція визначає, чи повинен нейрон, до якого вона прикріплена, бути активованим (увімкненим), на основі того, чи є вхідний сигнал нейрона релевантним для прогнозування моделі.

Існують різні математичні функції, які можуть використовуватися як активаційні функції, залежно від їх конкретного призначення. Лінійна активаційна функція обчислює вихідний сигнал нейрона шляхом множення ваг на входи згідно з рівнянням [див. Формулу (2.2)].

$$f(x) = x_1 \times w_1 + x_2 \times w_2 + x_3 \times w_3 \dots x_n \times w_n \quad (2.2)$$

де  $x$  – вхідний сигнал;

$w$  - значення ваг кожного нейрону.

Вихід лінійної активаційної функції змінюється від  $-\infty$  до  $+\infty$ , як показано на рис 2.6. Це означає, що лінійна активаційна функція така ж, як і відсутність активації.

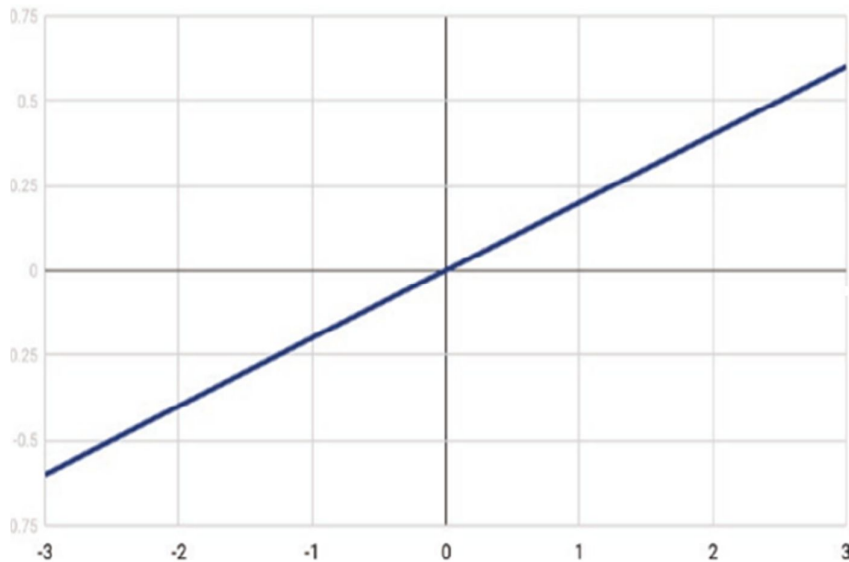


Рисунок 2.6 – Лінійна функція активації

Джерело: розроблено автором на основі [51]

Лінійна активаційна функція має такі дві основні проблеми:

1. Глибоке навчання використовує метод, який називається зворотним поширенням, який використовує техніку градієнтного спуску. Градієнтний спуск вимагає обчислення першої похідної від вхідного сигналу, яка у випадку лінійної активації є константою. Перша похідна константи дорівнює нулю. Це означає, що вона не має жодного зв'язку з вхідним сигналом. Отже, неможливо повернутися назад і оновити ваги вхідних нейронів.
2. Лінійна активаційна функція не може вирішити проблеми, які не є лінійно розділеними. Оскільки глибоке навчання зазвичай використовується для вирішення складних нелінійних завдань, лінійна активаційна функція не підходить для цього.

Сигмоїдна активаційна функція (рис 2.7), також відома як логістична активаційна функція, обчислює вихідний сигнал нейрона за допомогою сигмоїдної функції [див. Формулу (2.3)].

$$f(x) = \frac{1}{(1+e^{-x})} \quad (2.3)$$

де  $x$  – вхідний сигнал;

$e$  – основа натурального логарифма.

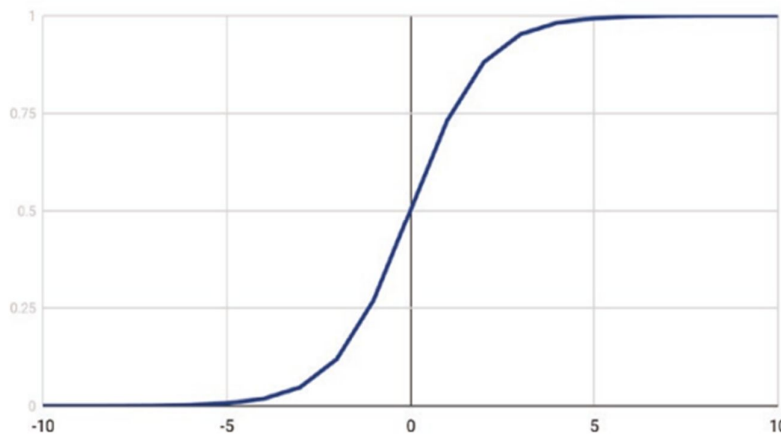


Рисунок 2.7 – Сигмоїдна функція активації

Джерело: розроблено автором на основі [51]

Сигмоїдна активаційна функція завжди дає значення між 0 та 1. Це робить вихідний сигнал плавним без стрибків, коли вхідне значення коливається. Іншою перевагою є те, що це нелінійна функція і не генерує константне значення від першої похідної. Це робить її придатною для глибокого навчання із зворотним поширенням, яке оновлює ваги на основі градієнтного спуску.

Найбільшим недоліком сигмоїдної активаційної функції є те, що вихідний сигнал не змінюється між великими або малими вхідними значеннями, що робить її непридатною для випадків, коли вектор ознак містить великі або малі значення. Один із способів подолати цей недолік - нормалізувати ваш вектор ознак, щоб він мав значення між -1 та 1 або між 0 та 1.

Гіперболічний тангенс (TanH), схожий на сигмоїдну активаційну функцію, за винятком того, що TanH є центрованим на нулі (рис. 2.8).

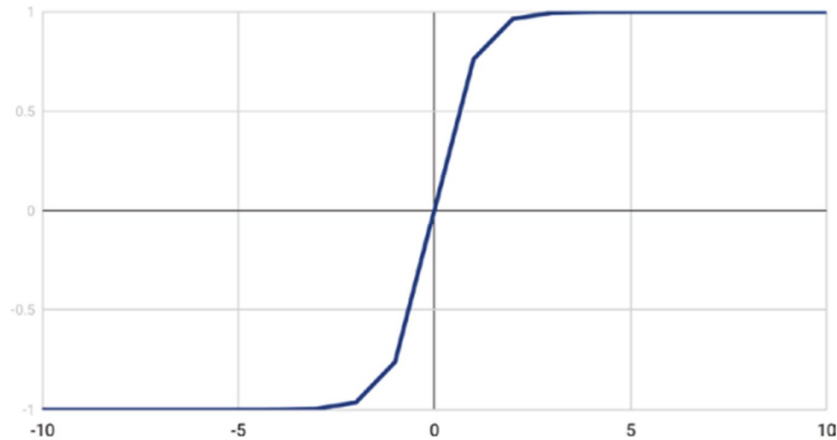


Рисунок 2.8 – Гіперболічний тангенс функція активації

*Джерело: розроблено автором на основі [51]*

Функція гіперболічного тангенса видає значення від -1 до +1 і має наступну вигляд [див. Формулу (2.4)].

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.4)$$

де  $x$  – вхідний сигнал;

$e$  – основа натурального логарифма.

Зрізаний лінійний вузол (ReLU) - функція, яка визначена таким чином [див. Формулу (2.5)]. Якщо значення  $x$  є додатнім, ReLU приймає це значення як вихід; інакше вона видає нуль. Вихідний сигнал ReLU знаходиться в діапазоні від 0 до  $+\infty$  (рис. 2.9).

$$f(x) = \max(0, x) \quad (2.5)$$

де  $x$  – вхідний сигнал;

ReLU є, найбільш популярною активаційною функцією для глибоких нейронних мереж.

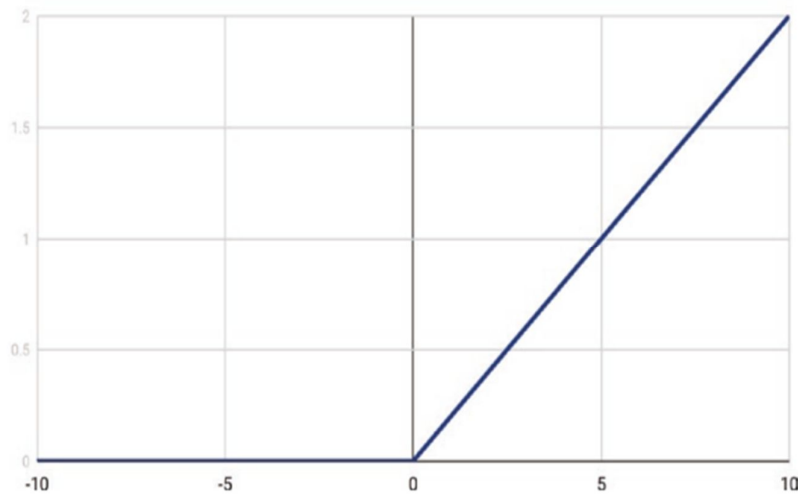
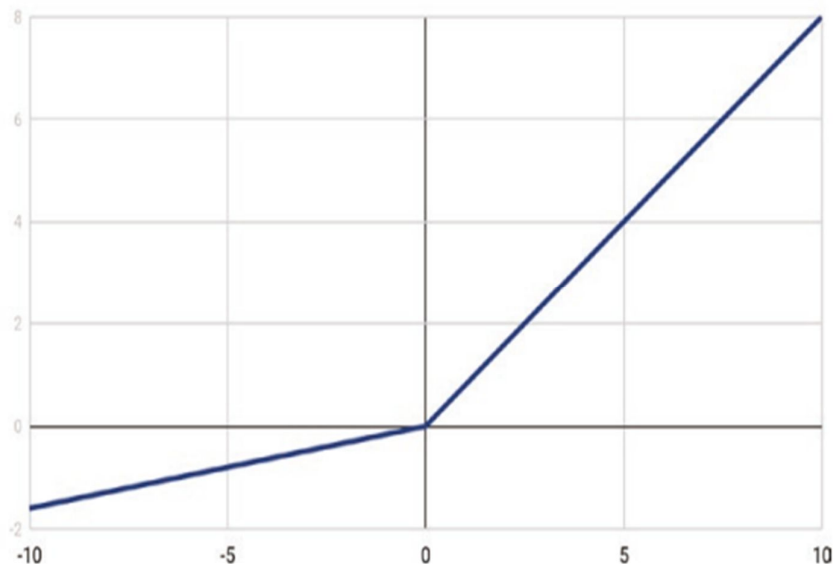


Рисунок 2.9 – ReLU функція активації

*Джерело: розроблено автором на основі [51]*

Перевагою активаційної функції ReLU є те, що вона обчислювально ефективна. Крім того, ReLU є нелінійною, і вона має похідну функцію, що робить її придатною для зворотного поширення для коригування ваг під час навчання нейронної мережі. Найбільшим недоліком функції ReLU є те, що градієнт функції стає нульовим для нульових або від'ємних входів. Це робить її непридатною для зворотного поширення, коли вхідний сигнал має від'ємні значення.

Leaky ReLU є варіантом функції ReLU, який вирішує проблему "мертвих" нейронів. Замість того, щоб перетворювати кожний від'ємний вхід на нуль, вона має невеликий нахил для від'ємного значення (рис. 2.10).



### Рисунок 2.10 – Leaky ReLU функція активації

*Джерело: розроблено автором на основі [51]*

Архітектура нейронної мережі визначає загальні принципи її побудови. Конфігурація конкретизує структуру мережі в рамках заданої архітектури: кількість нейронів, кількість входів та виходів мережі, використані активаційні функції. Структура штучної нейронної мережі являє собою орієнтований граф, в якості вузлів якого виступають нейрони, а в якості дуг - зв'язки з ваговими коефіцієнтами. Загалом, розрізняють наступні базові архітектури для задач регресії:

Мережі прямого поширення є одним з найпростіших та найширше використовуваних типів нейронних мереж. Характерною рисою таких мереж є те, що сигнал та дані в них поширюються в одному напрямку: від вхідних вузлів через приховані шари (якщо вони є) до вихідних вузлів, без будь-яких циклів чи зворотних зв'язків [52].

Багатошаровий перцептрон є найпоширенішим і найпростішим видом нейронних мереж прямого поширення. Багатошаровий перцептрон складається з одного (наскрізного) вхідного шару, одного або більшої кількості шарів, які називаються прихованими шарами (hidden layers), та ще одного шару елементів, який називається вихідним шаром (output layer), як показано на рис. 2.11.

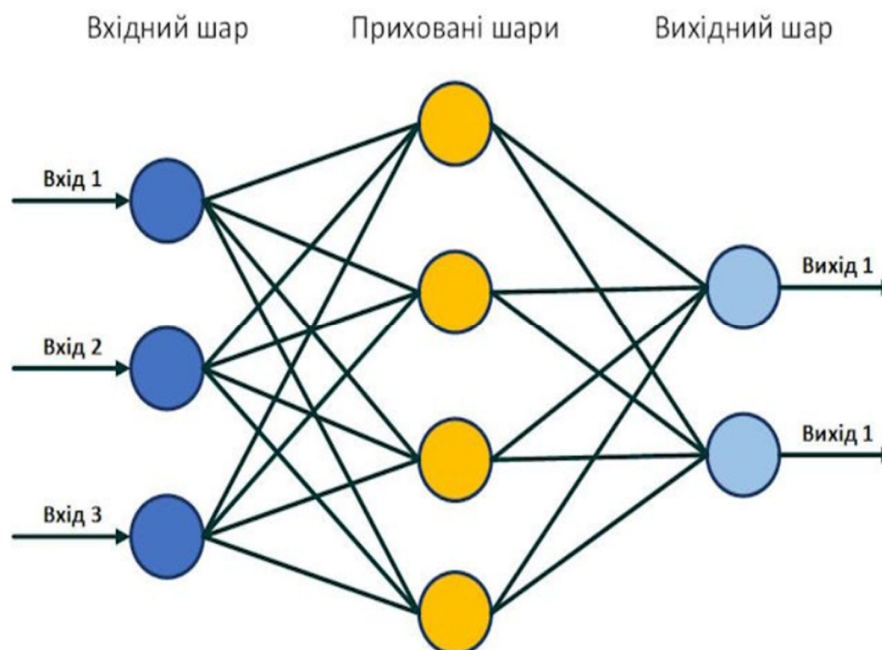


Рисунок 2.11 – Структура багатошарового перцептрону

*Джерело: розроблено автором на основі [53]*

Шари, розташовані ближче до вхідного шару, зазвичай називають нижчими шарами, а шари, що знаходяться ближче до виходів - вищими шарами. Кожен шар, крім вихідного, включає нейрон зміщення та повністю пов'язаний з наступним шаром. Коли нейронна мережа містить глибоку стопку прихованих шарів, вона називається глибокою нейронною мережею (deep neural network - DNN).

Об'єднуючи велику кількість таких нейронів, штучні нейронні мережі можуть вирішувати складні завдання, такі як кластеризація, класифікація, регресійний аналіз, комп'ютерний зір та машинний переклад. Особливостями штучних нейронних мереж є здатність до навчання та нелінійність, які дозволяють знаходити складні залежності між даними та враховувати досвід для покращення результату своєї роботи.

Вхідний вузол (input node) - це перший шар нейронної мережі, який приймає вхідні дані для обробки. Вхідні вузли не виконують жодних обчислень, вони просто передають вхідні значення до наступного прихованого шару мережі також вхідні вузли не мають вагових коефіцієнтів, ваги та зміщення присутні лише в прихованих шарах та вихідному шарі нейронної мережі. Кількість вхідних вузлів визначається розмірністю вхідних даних. Вхідні вузли можуть приймати дані різних типів, таких як числові значення, категоріальні дані або бінарні значення.

Приховані шари (hidden layers) є проміжними шарами нейронної мережі, розташованими між вхідним та вихідним шарами. Вони виконують основні обчислення та виявлення ознак у вхідних даних. Кожен прихований шар складається з певної кількості нейронів або вузлів. Ці нейрони отримують вхідні сигнали від попереднього шару (вхідного або прихованого), обробляють їх за допомогою активаційної функції та передають результати до наступного шару. Приховані шари відіграють ключову роль у здатності нейронної мережі вивчати складні нелінійні залежності у даних. Чим більше прихованих шарів і нейронів у мережі, тим більш складні функції вона може апроксимувати.

Вихідний шар (output layer) є останнім шаром нейронної мережі, який генерує кінцевий результат або прогноз на основі обробки вхідних даних через

приховані шари. Нейрони вихідного шару отримують вхідні сигнали від останнього прихованого шару та застосовують активаційну функцію для генерації остаточного результату. Кількість нейронів у вихідному шарі визначається типом задачі, яку вирішує нейронна мережа. Наприклад, для задач бінарної класифікації у вихідному шарі буде лише один нейрон, який генерує значення 0 або 1. Для задач багатокласової класифікації кількість нейронів дорівнюватиме кількості класів. У задачах регресії вихідний шар матиме один нейрон для безперервного цільового значення.

Рекурентні нейронні мережі – це клас нейронних мереж, що спеціально призначені для обробки послідовних даних, тобто даних, де порядок має значення. Вони використовуються в різних задачах обробки природної мови, часових рядів, генерації музики (рис 2.12).

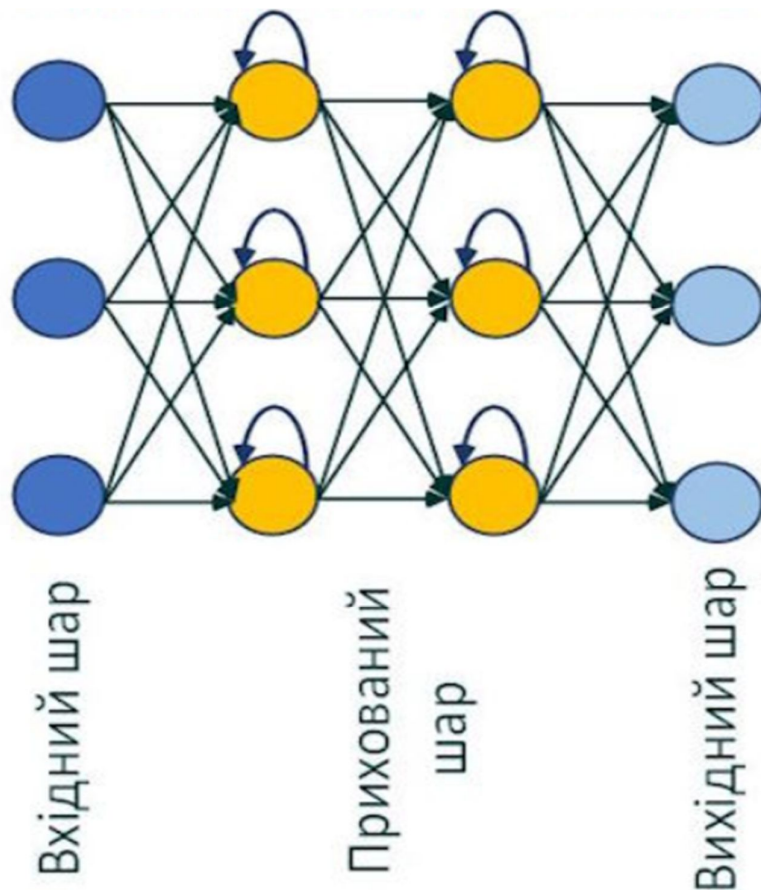


Рисунок 2.12 – Структура рекурентної нейронної мережі

*Джерело: розроблено автором на основі [53]*

Основні компоненти та принцип роботи:

1. Циклічні зв'язки: Відмінною рисою рекурентних нейронних мереж є циклічні (зворотні) зв'язки між нейронами, що дозволяють зберігати інформацію про попередні дані в послідовності. Це дозволяє мережі враховувати не тільки поточний вхід, але й контекст, заданий попередніми входами.
2. Вхідні та вихідні дані: В рекурентних нейронних мережах вхідні дані можуть подаватись як по одному елементу послідовності за раз, так і цілою послідовністю за один крок. Вихідні дані також можуть бути як одиничними реакціями на кожен вхід, так і однією відповіддю на всю послідовність.
3. Згортка пам'яті: Хоча RNN теоретично можуть зберігати інформацію про будь-які попередні входи в послідовності, на практиці вони часто стикаються з проблемою зникаючих або вибухаючих градієнтів, що ускладнює збереження інформації на довгі періоди.

Рекурентні нейронні мережі застосовуються в різних сферах. Вони ефективно використовуються для обробки та генерації мови, включаючи переклад та розпізнавання мови. В аналізі часових рядів RNN допомагають прогнозувати погоду та фінансові показники. Рекурентні нейронні мережі здатні моделювати комплексні залежності у послідовностях та адаптуватись до різних довжин вхідних даних, що робить їх надзвичайно потужним інструментом для аналізу послідовних даних.

Навчальна мета нейронної мережі полягає у визначенні найбільш оптимізованих ваг, за яких втрати є мінімальними. Коли мережа починає навчатися, вона присвоює ваги кожному вхідному зв'язку. Спочатку ці ваги рідко бувають оптимізованими. Те, наскільки ваги відхиляються від оптимізації, визначається шляхом вимірювання втрат (або помилки). Щоб визначити ідеальні ваги, алгоритм навчання оптимізує функцію втрат, щоб знайти ваги, які роблять функцію втрат мінімальною. Ваги оновлюються, і процес повторюється, доки немає більше можливості для оптимізації. Математична функція, яка оптимізує функцію втрат, називається алгоритмом оптимізації або оптимізатором.

Градiєнтний спуск - це алгоритм оптимізації, який знаходить ваги, за яких функція втрат дорівнює нулю або є мінімальною, отже градієнтний спуск - це техніка для знаходження мінімальної функції втрат [54].

Функція помилки представлена формулою (2.6):

$$f(w) = \frac{1}{N} \sum (y_i - w_i x_i) \quad (2.6)$$

де  $y_i$  – фактичне/відоме значення;

$w_i$  – вага, що відповідає вектору ознак  $x_i$  зразка.

Перша похідна функції в точці дає нахил або градієнт функції в цій точці. Похідна обчислюється для отримання градієнта, щоб визначити, в якому напрямку вздовж кривої рухатися, щоб отримати новий набір ваг. Оскільки метою є мінімізація вартості, алгоритм рухається в напрямку від'ємного градієнта [54].

Функція вартості матиме вигляд, як на лівому зображенні на рис. 2.13.



Рисунок 2.13 – Алгоритм роботи градієнтного спуску

Джерело: розроблено автором на основі [55]

1. Спочатку алгоритм обчислює вартість або втрати для початкових ваг, припускаючи, що ця втрата є  $f(w)$ , і припускаючи, що втрата обчислюється в точці 0 на (рис. 2.13).
2. Потім алгоритм обчислює градієнт і рухається вниз по кривій; напрямок визначається від'ємним градієнтом.

3. Під час спуску алгоритм обчислює нові ваги.
4. Помилка знову обчислюється з використанням нового значення ваги, і процес повторюється, доки алгоритм не знайде остаточну мінімальну вартість.

Коефіцієнт навчання визначає розмір кроків, які робить алгоритм під час спуску вздовж кривої в пошуках глобального мінімуму.

Вибір відповідного значення для коефіцієнта навчання є важливим. Якщо коефіцієнт навчання встановлено на велике значення, це може призвести до того, що алгоритм проскочить мінімальну точку, що призведе до коливань і неможливості збігтися. Навпаки, вибір малого коефіцієнта навчання потребуватиме великої кількості кроків для досягнення мінімальної точки, що призведе до повільної збіжності. Варто зазначити, що малий коефіцієнт навчання може значно сповільнити процес навчання. Для забезпечення ефективного навчання вкрай важливо встановити відповідний коефіцієнт навчання. Зазвичай практичний діапазон для коефіцієнта навчання знаходиться між 0,01 та 0,1.

Гرادієнтний спуск обчислює градієнти для всіх навчальних прикладів на кожному кроці та ітерації. Цей процес може бути обчислювально дорогим і займати багато часу, особливо для великих наборів даних. Стохастичний градієнтний спуск (SGD) вирішує ці проблеми [56].

SGD обчислює градієнти на основі невеликої підмножини навчального набору, яка легко вміщується в пам'ять. Отже, обирається випадковий набір вхідних даних, щоб усунути упередженість і забезпечити випадковість в оцінці градієнта.

Наступним кроком обчислюються градієнти, використовуючи випадково вибрану одну точку даних або невелику партію даних, та оновлюються ваги.

У SGD часто обчислюють оновлення ваг для кількох навчальних прикладів одночасно, а не для одного прикладу. Це допомагає зменшити дисперсію ваг і призводить до більш стабільної збіжності.

Використовуючи SGD, обчислювальні витрати значно зменшуються, що робить можливим навчання моделей на великих наборах даних. Використання менших підмножин також вносить випадковість і може допомогти алгоритму уникнути локальних мінімумів та дослідити різні частини простору оптимізації.

AdaGrad (Adaptive Gradient) - це алгоритм оптимізації, який автоматично адаптує коефіцієнти навчання для різних параметрів під час навчання нейронної мережі. У звичайному градієнтному спуску використовується єдиний глобальний коефіцієнт навчання для всіх параметрів. Однак різні параметри можуть мати різні діапазони частот оновлення, тому фіксований коефіцієнт навчання може бути неоптимальним. AdaGrad адаптує коефіцієнт навчання для кожного параметра окремо, присвоюючи більші коефіцієнти рідкісним параметрам (з невеликими градієнтами) і менші коефіцієнти часто оновлюваним параметрам (з великими градієнтами). AdaGrad дуже ефективний у роботі з розрідженими даними, такими як в комп'ютерному зорі або обробці природної мови (NLP) [57].

### **2.3 Автоматизоване машинне навчання для побудови нейронних мереж**

Кожна система машинного навчання має гіперпараметри, і основним завданням в автоматизованому машинному навчанні є автоматичне налаштування цих гіперпараметрів для оптимізації продуктивності. Сучасні глибокі нейронні мережі особливо сильно залежать від широкого спектру гіперпараметрів, які визначають архітектуру нейронної мережі, регуляризацію та оптимізацію.

Автоматизована оптимізація гіперпараметрів має кілька важливих застосувань і дозволяє скоротити людські трудовитрати, необхідні для застосування машинного навчання. Також автоматизоване машинне навчання може покращити продуктивність алгоритмів машинного навчання (адаптуючи їх до конкретного завдання). Автоматизований процес явно більш відтворюваний, ніж ручний пошук, він полегшує справедливе порівняння наукових результатів, оскільки різні методи можна справедливо порівнювати лише в тому випадку, якщо всі вони мають однаковий рівень налаштування для вирішення поставленого завдання [58].

Однак, реалізація AutoML стикається з кількома проблемами, які ускладнюють застосування технології на практиці. Оцінка функцій для великих моделей або великих наборів даних виходить надзвичайно дорого, також простір

конфігурацій часто буває складним (що складається з комбінації безперервних, категоріальних і умовних гіперпараметрів) і багатовимірним. Крім того, не завжди зрозуміло, які з гіперпараметрів алгоритму необхідно оптимізувати і в яких діапазонах.

Пошук по сітці (grid search) - це найбазовіший метод, також відомий як повне факторне планування (full factorial design). Користувач задає скінченну множину значень для кожного гіперпараметра, а пошук по сітці оцінює декартовий добуток цих множин. Такий підхід страждає від прокляття розмірності, оскільки необхідна кількість оцінок функцій зростає експоненціально в міру збільшення розмірності. Додатковим недоліком пошуку по сітці є те, що збільшення роздільної здатності дискретизації (зменшення кроку гіперпараметрів) істотно збільшує необхідну кількість оцінок функцій.

Простою альтернативою пошуку по сітці є випадковий пошук (random search). Випадковий пошук вибирає конфігурації випадковим чином, поки не буде вичерпаний певний ресурс на пошук. Цей підхід працює краще, ніж пошук по сітці, коли деякі гіперпараметри набагато важливіші за інші. Випадковий пошук є корисною базовою схемою, оскільки він не робить жодних припущень щодо оптимізованого алгоритму машинного навчання і – за достатньої кількості ресурсів – прагне досягти продуктивності, доволіно близької до оптимальної. Тому чергування випадкового пошуку з більш складними стратегіями оптимізації дозволяє гарантувати мінімальну швидкість збіжності. Випадковий пошук також є корисним методом для ініціалізації процесу пошуку, оскільки він досліджує весь простір конфігурації і, таким чином, часто знаходить параметри з прийнятною продуктивністю.

Баєсівська оптимізація є потужним методом, який використовується для ефективного пошуку глобального оптимуму складних функцій, що мають багато вхідних параметрів. Баєсівська оптимізація ґрунтується на принципах баєсівської статистики та використовує апріорну інформацію про функцію, яку потрібно оптимізувати, у поєднанні з результатами попередніх спостережень (оцінок функції в певних точках). На основі цієї інформації будується апроксимація функції, яка

називається сурогатною моделлю або моделлю заміщення (surrogate model). Сурогатна модель використовується для прогнозування значень функції в нових точках, а також для оцінки невизначеності цих прогнозів.

Процес баєсівської оптимізації складається з кількох ітерацій, на кожній з яких виконуються наступні кроки:

1. Побудова сурогатної моделі на основі наявних спостережень функції.
2. Використання сурогатної моделі для визначення найбільш перспективної точки для наступного спостереження функції.
3. Обчислення значення функції в обраній точці та додавання цього спостереження до набору даних.
4. Повторення попередніх кроків до досягнення критерію зупинки (наприклад, максимальної кількості ітерацій або заданої точності).

Баєсівська оптимізація може знаходити глобальний оптимум складних функцій з меншою кількістю обчислень, ніж традиційні методи, такі як випадковий пошук або сітковий пошук.

Пошук нейронної архітектури (neural architecture search, NAS), що являє собою процес автоматизації проектування архітектури, є логічним кроком в автоматизації машинного навчання, а отже, NAS можна розглядати як область AutoML.

Класифікація методів NAS проводиться за трьома параметрами: простором пошуку, стратегією пошуку та стратегією оцінки продуктивності.

Простір пошуку - визначає, які архітектури можуть бути представлені в принципі. Наявність попередніх знань про властивості, які добре підходять для виконання завдання, може зменшити розмір простору пошуку і спростити пошук. Простір пошуку визначається такими параметрами, як: максимальне число шарів  $p$ , тип операції, яку може виконувати кожен шар, і гіперпараметри, пов'язані з операцією, наприклад, число фільтрів, розмір ядра або просто число блоків для повнозв'язних мереж.

Стратегія пошуку - описує, як досліджувати простір пошуку. Вона являє собою класичний компроміс між дослідженням і використанням, оскільки, з одного

боку, бажано швидко знайти добре працюючі архітектури, а з іншого боку, слід уникати передчасної збіжності в область субоптимальних архітектур. Для дослідження простору нейронних архітектур можна використовувати безліч різних стратегій пошуку, включаючи випадковий пошук, байєсівську оптимізацію і градієнтні методи.

Стратегія оцінки продуктивності - метою NAS зазвичай є пошук архітектур, які мають високу передбачувальну здатність на невидимих даних. Оцінка продуктивності відноситься до процесу оцінки цієї продуктивності: найпростішим варіантом є стандартне навчання і перевірка архітектури на даних [59].

AutoKeras є відкритою бібліотекою автоматизованого машинного навчання для нейронних мереж, розробленою компанією DATA Lab при Техаському університеті A&M. Вона базується на популярній бібліотеці TensorFlow і спрямована на спрощення та прискорення процесу створення та налаштування нейронних мереж для різноманітних задач. Основна ідея AutoKeras полягає в тому, щоб автоматизувати процес пошуку архітектури нейронної мережі, налаштування гіперпараметрів та вибору оптимізаторів, зменшуючи потребу в ручному налаштуванні та експериментуванні. Бібліотека використовує різні методи автоматизованого машинного навчання, такі як байєсівська оптимізація та нейроеволюційні алгоритми, для ефективного пошуку оптимальних конфігурацій моделі [60].

AutoKeras дозволяє викорисовувати автоматичний пошук архітектури нейронної мережі. Ця процедура включає вибір типів шарів, кількості шарів, розмірів шарів, функцій активації та з'єднань між шарами. AutoKeras автоматично тренує та валідує ці архітектури, намагаючись знайти найкращу конфігурацію для заданої задачі.

Після визначення архітектури нейронної мережі AutoKeras автоматично налаштовує гіперпараметри моделі, такі як коефіцієнт навчання, регуляризація, розмір батчу та інші. Це робиться за допомогою методів оптимізації, таких як випадковий пошук або байєсівська оптимізація, з метою максимізації продуктивності моделі. Крім того, AutoKeras автоматично вибирає та налаштовує

оптимізатори для навчання нейронної мережі. Бібліотека може випробувати різні оптимізатори, такі як SGD, Adam, RMSProp, та налаштувати їх параметри для досягнення швидшої збіжності та кращої продуктивності.

Після завершення процесу автоматизованого пошуку архітектури, налаштування гіперпараметрів та вибору оптимізаторів, AutoKeras тренує остаточну модель на повному наборі даних. Бібліотека також надає інструменти для оцінки продуктивності моделі, такі як метрики точності, матриця помилок та візуалізація результатів.

Одна з головних переваг AutoKeras полягає в її простоті використання. Це робить AutoKeras привабливим інструментом для роботи.

Однак слід зазначити, що AutoKeras, як і інші інструменти AutoML, має певні обмеження. Наприклад, процес автоматизованого пошуку архітектури може бути обчислювально дорогим, особливо для великих наборів даних та складних задач. Незважаючи на ці обмеження, AutoKeras є потужним інструментом, який спрощує та прискорює процес створення нейронних мереж, роблячи його більш доступним для широкого кола користувачів. Бібліотека продовжує активно розвиватися та вдосконалюватися, пропонуючи нові функції та можливості для автоматизації машинного навчання.

### **РОЗДІЛ 3**

## **ПРАКТИЧНА РЕАЛІЗАЦІЯ DATA DRIVEN ПІДХОДУ ТА МОДЕЛЬНОГО ІНСТРУМЕНТАРІЮ ДЛЯ ВИБОРУ ОБ'ЄКТІВ ЖИТЛОВОГО ІНВЕСТИВАННЯ**

### **3.1 Результати препроцесингу структурованих та неструктурованих даних для моделювання ціни на об'єкти нерухомості**

Для реалізації та навчання моделей нейронної мережі необхідно підготувати набір даних, на основі якого буде відбуватися налаштування вагових коефіцієнтів. Оскільки моделі машинного навчання на основі нейромереж потребують великого обсягу даних, то було зібрано набір даних, що складається з 5000 об'єктів вторинної нерухомості, кожен з яких містить 16 характеристик об'єкту та цільову змінну ціни об'єкта нерухомості. Дані були зібрані у період з серпня по листопад 2024 року, та містить як числові, текстові, так і категоріальні змінні.

Зібрані дані мають наступну структуру: Числові значення: площа об'єкту нерухомості, площа кухні, кількість житлових кімнат, кількість ванних кімнат, поверх, на якому розташований об'єкт, а також кількість поверхів у будинку.

Категоріальні значення: район розташування об'єкту нерухомості в місті Київ, тип стін, з яких побудований будинок об'єкту нерухомості, наявність меблювання, вид опалення, клас будинку, планування кімнат об'єкту, тип та наявність ремонту, тип будинку, в якому розташована квартира.

Текстові дані: опис об'єкту нерухомості, наявність комунікацій у квартирі, вид побутової техніки, наявність мультимедійної техніки.

Для збору даних була розроблена програма на мові Python, яка збирає обрану користувачем інформацію по кожному об'єкту нерухомості та зберігає її у вихідному файлі. Програма-парсер переглядає сторінки із веб-ресурсу, зберігає вихідні посилання на кожен об'єкт нерухомості та екстрагує потрібні дані з вихідного коду кожного з посилань.

Для реалізації обробки даних, а також побудови моделей з використанням нейронних мереж, у роботі використовується мова програмування Python та бібліотеки цієї мови програмування. Створення коректного набору даних включає збір даних, очищення даних, застосування фільтрації, а також трансформацій, для того щоб непідготовлені дані перетворилися на вид, який буде підходити для коректної роботи моделей нейронних мереж.

У процесі підготовки даних для використання в нейронній мережі методологічний підхід передбачає кілька важливих етапів. Важливість даних обумовлена тим, що якість роботи нейронної мережі в значній мірі залежить від характеристик підготовленого набору даних.

Перший етап - збір даних. Під час цього етапу відбувається збір наявних даних для досягнення поставленої дослідницької цілі, а саме прогнозування ціни на об'єкт нерухомості. Наступний етап - очищення даних. Під очищенням даних розуміється ряд специфічних операцій:

1. Видалення дублікатів. Дубльовані дані можуть спричинити спотворення результатів дослідження за рахунок повторення однакових спостережень. Це може призвести до збільшення об'єму даних без збільшення корисної інформації, збільшуючи обчислювальні витрати та схильність моделі до перенавчання.
2. Видалення або заміна пропущених значень. Відсутні дані можуть заповнюватися за допомогою різних технік, таких як використання середніх значень, медіан або моди, або ж дані з пропусками можуть бути повністю видалені з набору даних.
3. Обробка аномалій. Аномалії відносяться до значень, які суттєво відрізняються від більшості даних. Вони вимагають ретельного аналізу та однозначного рішення: їх або видаляють, або коригують, щоб зробити результати більш точними.

Після етапу очищення даних настає етап перетворення:

1. Нормалізація та стандартизація даних. Ці процеси полягають у приведенні даних до єдиної шкали. Нормалізація, як правило, полягає в приведенні значень до діапазону від 0 до 1, тоді як стандартизація приводить дані до розподілу зі середнім значенням 0 і стандартним відхиленням 1.
2. Перетворення категоріальних даних. Багато моделей машинного навчання обробляють лише числові дані, що вимагає необхідності перетворення категоріальних значень на числові. Існують різні техніки, такі як One-Hot Encoding або Label Encoding. В даній роботі будуть використані обидві техніки перетворення категоріальних даних в числові, з урахування доцільності використання кожної з технік.

Щоб виконати завдання моделювання цін на об'єкти нерухомості, спершу маємо встановити та імпортувати усі потрібні пакети. Для цього використовується спеціальна команда інсталяції необхідних бібліотек, а також спеціальна функція котра дає можливість безпосереднього використання встановлених бібліотек у коді.

Обробка пропущених значень за категоріальними стовпцями буде проводитися за допомогою заміни пропущеного значення на моду за кожним зі стовпців.

Після заповнення усіх пропущених категоріальних даних слід застосувати техніку кодування міток.

Техніка кодування міток (англ. Label Encoding) є методикою перетворення категоріальних значень в числовий формат для подальшого їх використання в моделях машинного навчання [61].

Цей підхід базується на принципі присвоєння кожному унікальному значенню всередині ознаки унікального числового ідентифікатора або мітки. Для побудови правильної моделі кодування категоріальних даних в числові мітки, була використана техніка знаходження середньої ціни за кожною з міток для кожної ознаки. Тобто, мінімальне значення 1 - присвоювалося типу мітки всередині ознаки, яка має мінімальне середнє значення цільової змінної ціни об'єкта, а максимальне

значення присвоювалося типу мітки всередині ознаки, яка має максимальну середню ціну.

Приклад використання техніки кодування міток представлено у табл. 3.1, для ознаки клас будинку, в якому розташовано об'єкт нерухомості.

Таблиця 3.1 - Приклад використання техніки кодування міток

| Клас будинку         | Середня ціна об'єкта нерухомості | Відповідна категоріальні мітка |
|----------------------|----------------------------------|--------------------------------|
| Економ клас будинку  | 2.5 млн. Грн.                    | 1                              |
| Комфорт клас будинку | 3.7 млн. Грн.                    | 2                              |
| Бізнес клас будинку  | 4.8 млн. Грн.                    | 3                              |
| Еліт клас будинку    | 6.6 млн. Грн.                    | 4                              |

*Джерело: розроблено автором на основі [61]*

Таким чином було перекодовано усі категоріальні фактори котрі були наведені в частині опису набору даних.

Для обробки текстових стовпців датасету, а саме: наявність комунікацій у квартирі, вид побутової техніки, вид та наявність мультимедійної техніки, було використано іншу техніку кодування а саме одночасне кодування (англ. One-hot Encoding).

Одночасне кодування - це підхід до обробки категоріальних даних в машинному навчанні. One-hot Encoding - це метод перетворення категоріальних змінних на бінарний формат. Він створює нові бінарні стовпці (0 і 1) для кожної категорії в оригінальній змінній. Кожна категорія в оригінальному стовпці представляється як окремий стовець, де значення 1 вказує на присутність цієї категорії, а 0 вказує на її відсутність [62].

Основна мета One-hot Encoding полягає в тому, щоб забезпечити ефективне використання категоріальних даних в моделях машинного навчання, надаючи більш детальне представлення категоріальних змінних, одночасне кодування може допомогти покращити продуктивність моделей машинного навчання. Ця техніка дозволяє моделям виявляти складні зв'язки в даних, які можуть бути пропущені, якщо категоріальні змінні були розглянуті як єдині сутності.

Для імплементації одночасного кодування, було використано бібліотеку scikit-learn, а такж вбудовану функцію CountVectorizer.

Реалізація одночасного кодування наведено на рис. 3.1.

```
# One hot encoding можливих значень категоріальних стовпців
for col in ['comfort', 'multimedia', 'utilities', 'household_appliances']:
    cv = CountVectorizer(tokenizer=lambda x: x.split(','))
    new_features = pd.DataFrame(cv.fit_transform(df[col]).toarray(), columns=cv.get_feature_names_out(), index=df.index)
    df_text_feature = pd.concat([df_text_feature, new_features], axis=1)
```

Рисунок 3.1 – Реалізація одночасного кодування факторів

Джерело: розроблено автором на основі [62]

В цій реалізації за допомогою циклу, котрий проходить по текстовим стовпцям, виконується почергова обробка стовпців, за допомогою функції CountVectorizer, яка повертає матрицю, де кожен стовпець відповідає унікальному значенню в оригінальному стовпці, а рядки відповідають записам в датафреймі. Значення в матриці – це кількість входжень кожного унікального значення в відповідний запис.

Приклад отриманих результатів одночасного кодування для стовпця наявність побутової техніки наведено в (табл. 3.2) та (табл. 3.3).

Таблиця 3.2 - Приклад вхідних даних для техніки одночасного кодування

| Об'єкт   | Наявність побутової техніки                                   |
|----------|---------------------------------------------------------------|
| Об'єкт 1 | Холодильник, Пральна машина                                   |
| Об'єкт 2 | Посудомийна машина, Холодильник, Варильна панель, Духова шафа |

Джерело: розроблено автором на основі [62]

Таблиця 3.3 - Приклад вихідних даних після застосування техніки одночасного кодування

| Об'єкт   | Холодильник | Пральна машина | Посудомийна машина | Варильна панель | Духова шафа |
|----------|-------------|----------------|--------------------|-----------------|-------------|
| Об'єкт 1 | 1           | 1              | 0                  | 0               | 0           |
| Об'єкт 2 | 1           | 0              | 1                  | 1               | 1           |

Джерело: розроблено автором на основі [62]

Для обробки стовпця опису об'єкта нерухомості, потрібно використовувати певний препроцесинг цього тексту. Оскільки опис є вільно сформульованим

текстом, який містить важливі деталі про об'єкт, він може включати різні формати, помилки, аббревіатури, стоп-слова та інші елементи, які можуть заважати аналізу тексту.

Перший крок перед обробкою опису, повинен бути очищення тексту. Це важливий етап, оскільки він допомагає підготувати дані для подальшого аналізу та обробки. Потрібно видалити зайві символи, такі як розділові знаки, цифри, спеціальні символи тощо, використовуючи регулярні вирази. Регулярні вирази - це інструмент для пошуку та маніпуляції текстовими даними за допомогою спеціальних шаблонів.

Крім того, нам необхідно видалити стоп-слова, які не несуть жодного семантичного значення для аналізу тексту. Стоп-слова - це найпоширеніші слова в мові, такі як артиклі, прийменники, сполучники. Для цього можна використати список стоп-слів з бібліотеки NLTK (Natural Language Toolkit), яка є бібліотекою для обробки природної мови в Python [63].

Після очищення тексту від зайвих символів та стоп-слів, застосовуємо техніку лемінгу. Лемінг - це процес приведення слова до його лемми (нормальної форми), який допомагає стандартизувати тексти і спрямовує увагу на основну форму слів. Лемітизація враховує контекст і граматичні правила мови, на відміну від стемінгу, який є більш грубим підходом до вилучення суфіксів. Це допомагає зменшити розмірність даних та усунути надлишковість, що може покращити продуктивність моделей обробки природної мови, оскільки різні форми одного і того ж слова будуть представлені єдиною леммою.

Після етапу лемітизації, використаю NLP (Natural Language Processing) техніку, таку як векторизація тексту, для конвертації тексту в числові дані. Векторизація тексту - це процес перетворення текстових даних у векторні представлення, які можуть бути використані для машинного навчання та інших задач обробки природної мови [64].

Нижче наведено програмний код реалізації техніки обробки текстового опису об'єкта нерухомості (рис. 3.2). А саме:

1. Видалення стоп слів.

2. Видалення символів які не є українськими літерами, використовуючи регулярний вираз.
3. Видалення слів що складаються з 1-6 символів, це корисно, оскільки ймовірно короткі слова не матимуть великого значення для аналізу.
4. Перетворення тексту у нижній регістр

```
# Завантаження бібліотеки
nltk.download('stopwords')
stop_words_uk = set(stopwords.words('ukrainian'))

# Видалення стоп слів
df['full_description'] = df['full_description'].apply(lambda x: ' '.join([word for word in x.split() if word not in stop_words_uk]))

# Видалення символів які не є українськими
df['full_description'] = df['full_description'].apply(lambda x: re.sub(r'^а-яА-ЯіІїєЄ\s', '', x, flags=re.IGNORECASE))

# Видалення слів що мають менше 6 символів
df['full_description'] = df['full_description'].apply(lambda x: re.sub(r'\b\w{1,6}\b', '', x, flags=re.IGNORECASE))

# Перетворення тексту до нижнього регістру
df['full_description'] = df['full_description'].str.lower()
```

Рисунок 3.2 – Реалізація обробки текстового фактору

*Джерело: розроблено автором на основі [64]*

Для того щоб застосувати лемінг, для текстового опису даних, використаю екземпляр класу “MorphAnalyzer” з бібліотеки “руmorphy2”. Програмна реалізація наведена на (рис 3.3).

```
import руморфy2

morph_uk = руморфy2.MorphAnalyzer(lang='uk')

# Застосування лемітизації до стовпця
df['full_description'] = df['full_description'].apply(lambda x: ' '.join([morph_uk.parse(word)[0].normal_form for word in x.split()])))
```

Рисунок 3.3 – Застосування лемінгу

*Джерело: розроблено автором на основі [64]*

Для визначення важливості слів в обробленому текстовому описі об’єкта нерухомості, та конвертації цих значень в числовий формат, застосую техніку TF-IDF (Term Frequency-Inverse Document Frequency). Ця техніка є однією з найпопулярніших технік для векторизації тексту. Вона дозволяє визначити важливість слів у тексті, враховуючи як їхню частоту появи в датасеті, так і їхню рідкість у всьому датасеті [65].

Ця техніка складається з двох компонентів: TF (Term Frequency) та IDF (Inverse Document Frequency).

TF (Term Frequency) - частота появи слова в документі: TF вимірює, наскільки часто слово з’являється в конкретному документі. Чим частіше слово зустрічається

в документі, тим більше його значення TF. Вираховується за формулою [див. Формулу (3.1)]:

$$tf(t, d) = \frac{n_t}{\sum_k n_k} \quad (3.1)$$

де  $n_t$  – кількість входжень слова  $t$  документ;

$\sum_k n_k$  – загальна кількість слів в документі.

IDF (Inverse Document Frequency) - зворотна частота документів: IDF вимірює, наскільки рідкісним є слово в корпусі документів. Чим рідше слово зустрічається в корпусі, тим більше його значення IDF. Вираховується за формулою [див. Формулу (3.2)]:

$$IDF = \log \frac{|D|}{|(d_i \supset t_i)|} \quad (3.2)$$

де  $|D|$  – кількість документів колекції;

$|(d_i \supset t_i)|$  – кількість документів, в яких зустрічається слово  $t_i$  ( $t_i \neq 0$ ).

Програмна реалізація наведена на (рис. 3.4).

```
# TF IDF техніка важливості слів в стопці опису об'єкта нерухомості
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
texts_tfidf = tfidf_vectorizer.fit_transform(df['full_description'])
```

Рисунок 3.4 – Застосування техніки TF-IDF

Джерело: розроблено автором на основі [66]

Наступним кроком для обробки текстового фактору є використання техніки “bag-of-words” (рис. 3.5). Ця техніка є одним з найпопулярніших та найпростіших підходів до представлення текстових даних у числовому вигляді для подальшого машинного навчання та аналізу. Основна ідея алгоритму “торба слів” полягає в тому, що текст розглядається як неупорядкований набір слів (або токенів), ігноруючи граматику та порядок слів у реченнях. Кожен унікальний термін у корпусі документів представляється як окрема змінна, а значенням цієї змінної для кожного документа є кількість входжень цього терміну в даний документ. Таким

чином, кожен документ перетворюється на вектор, де кожен елемент відповідає певному терміну, а його значення - це кількість входжень цього терміну в документі [65].

```
words = tfidf_vectorizer.get_feature_names_out()
bag_of_words = pd.DataFrame(texts_tfidf.todense(), columns=words)
```

Рисунок 3.5 – Застосування техніки “bag-of-words”

*Джерело: розроблено автором на основі [65]*

Після використання техніки “bag-of-words” для перетворення текстових даних на числовий формат, наступним кроком застосую метод не негативної матричної факторизації (Non-negative Matrix Factorization, NMF).

NNMF — це алгоритм розкладання матриці, який використовується для зменшення розмірності, основна відмінність від інших методів полягає в тому, що NMF тримає всі елементи матриці невід'ємними. NMF створює дві нові матриці з меншим розміром з поданих даних. Процес полягає в пошуку двох матриць, які при їх перемноженні найближче відповідають початковій матриці (рис. 3.6) [67].

```
from sklearn.decomposition import NMF
nmf = NMF(n_components=10)
nmf_result = nmf.fit_transform(bag_of_words)
```

Рисунок 3.6 – Застосування не негативної матричної факторизації

*Джерело: розроблено автором на основі [67]*

Підсумковим етапом реалізації процесу обробки текстового фактору є застосування алгоритму відображення UMAP (Uniform Manifold Approximation and Projection) до даних, отриманих після застосування NMF. Концептуально, UMAP базується на встановленні зв'язків між високорозмірними та низькорозмірними просторами. Цей алгоритм шукає точки в більш низькорозмірному просторі, що відтворюють структуру високорозмірних точок якомога точніше, зазначаючи, що відстані між точками нижчої розмірності мають якнайбільше наближатися до відстаней між високорозмірними точками. Алгоритм починається з конструювання графа з вхідних точок з високою розмірністю, який відображає структуру даних. Згодом, він оптимізує місця вихідних точок, розташованих у низькому просторі так, щоб вони дотримувались структури графа якомога краще (рис. 3.7) [68].

```
from umap import UMAP
umap = UMAP(n_components=2)
umap_result = umap.fit_transform(nnmf_result)
umap_df = pd.DataFrame(umap_result, columns=['dim_1', 'dim_2'])
```

Рисунок 3.7 – Застосування техніки UMAP

*Джерело: розроблено автором на основі [68]*

### 3.2 Аналіз результатів нейромережевого моделювання та прогнозування ціни на житлову нерухомість

Одним з ключових етапів у процесі створення ефективної моделі машинного навчання є ретельний аналіз вхідних даних. Особливу увагу під час аналізу вхідних даних слід приділити цільовій змінній, адже саме її значення ми намагаємося передбачити або пояснити за допомогою моделі.

На початковому етапі аналізу цільової змінної ціни необхідно провести дослідження її статистичних характеристик, таких як середнє значення, діапазон значень. Ці показники дозволять нам зрозуміти загальну структуру розподілу цін, виявити можливі викиди у наборі даних.

Для візуалізації розподілу цільової змінної ціни та виявлення можливих викидів буде використано коробковий графік. Цей тип графіка є оптимальним вибором для аналізу розподілу даних, оскільки він надає стислу, але інформативну візуалізацію ключових статистичних показників, таких як медіана, нижній і верхній квартилі, а також мінімальні та максимальні значення. Крім того, ящикова діаграма дозволяє легко ідентифікувати викиди - значення, що значно відхиляються від основного розподілу даних (рис. 3.8).

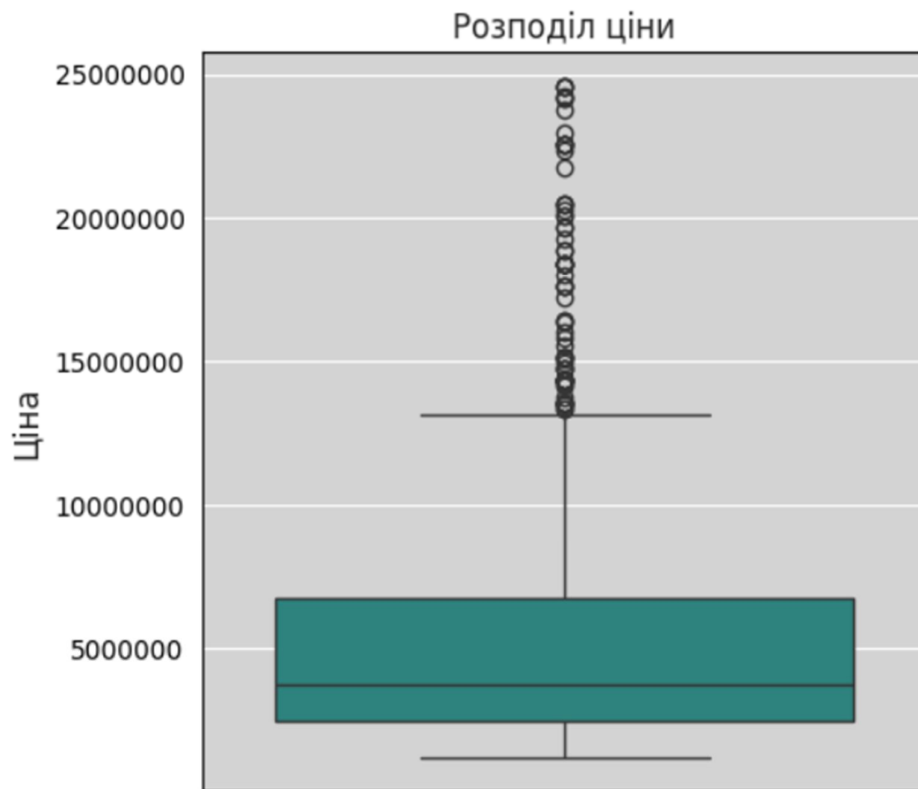


Рисунок 3.8 – Аналіз цільової змінної ціни об'єкта нерухомості

*Джерело: розроблено автором на основі [69]*

За результатами ящикової діаграми для цільової змінної ціни, ми можемо зробити наступні висновки:

Середня ціна у наборі даних становить 5,5 млн. грн.. Мінімальна ціна у наборі даних дорівнює 1,22 млн. грн., що відображається нижнім "вусом" на діаграмі. Водночас, максимальна ціна сягає 24,6 млн. грн.

Проте, на діаграмі ми спостерігаємо кілька точок, які значно відхиляються від основного розподілу цін. Ці точки, відомі як викиди, можуть спотворювати результати аналізу та негативно впливати на точність моделі. Тому, з метою підвищення якості даних та покращення продуктивності моделі, було видалено об'єкти ціна котрих перевищує 13 млн. грн.

Крім того, форма прямокутника на діаграмі вказує на те, що розподіл цін не є оптимальним. Ідеальним вважається симетричний розподіл, коли медіана розташована посередині прямокутника, а відстані від медіани до нижнього та верхнього кватилів приблизно однакові. Однак, у моєму випадку, спостерігається асиметричний розподіл, що може свідчити про наявність зміщення у даних.

Також потрібно проаналізувати площу об'єктів нерухомості (рис. 3.9).

Для цього побудую також коробчасту діаграму.

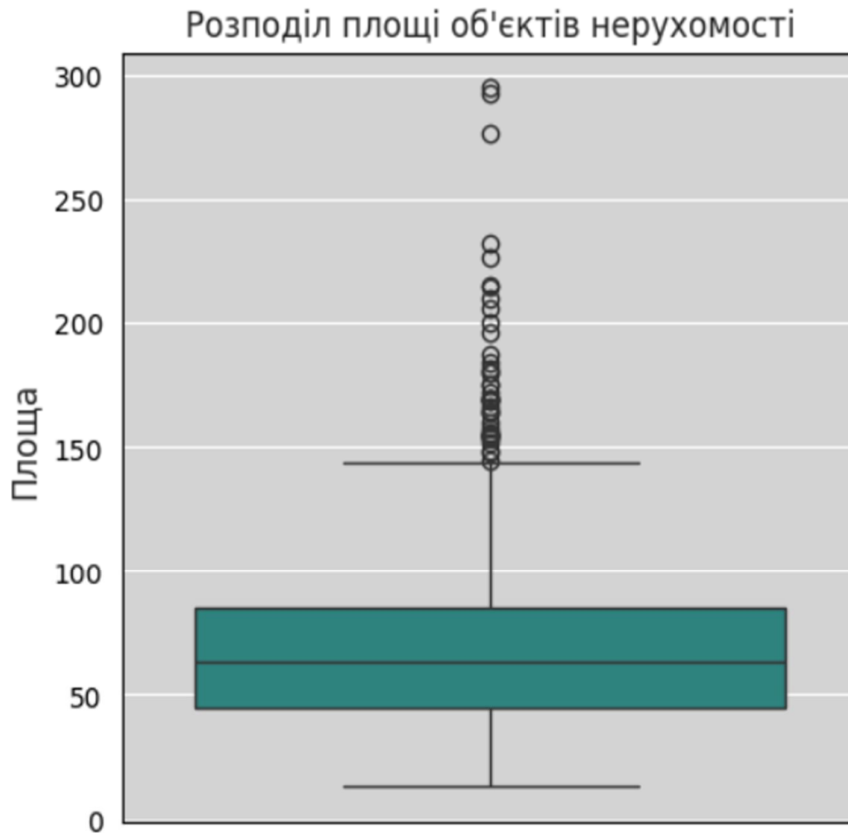


Рисунок 3.9 – Аналіз розподілу площі об'єктів нерухомості

*Джерело: розроблено автором на основі [69]*

За результатами візуалізації видно, що середня площа об'єктів становить 71 метр квадратний, мінімальна площа становить 13 метрів квадратних - це значення є досить низьким і може відповідати дуже маленьким квартирам.

Нижнє та верхнє "вуса" діаграми відображають діапазон значень, які не вважаються викидами. Однак, в моєму наборі даних, спостерігається кілька точок, розташованих вище верхнього "вуса", тому видалю також об'єкти де площа більше 150 метрів квадратних.

Наступним кроком буде побудова кореляційної матриці (рис. 3.10). Кореляційна матриця представляє собою таблицю, в якій кожен елемент відображає коефіцієнт кореляції між двома змінними. Коефіцієнт кореляції може набувати значень від -1 до 1, де -1 вказує на сильний негативний зв'язок, 0 – на відсутність зв'язку, а 1 – на сильний позитивний зв'язок. Аналізуючи кореляційну матрицю для

даних про нерухомість, можна виявити, які характеристики мають найбільший вплив на ціну.

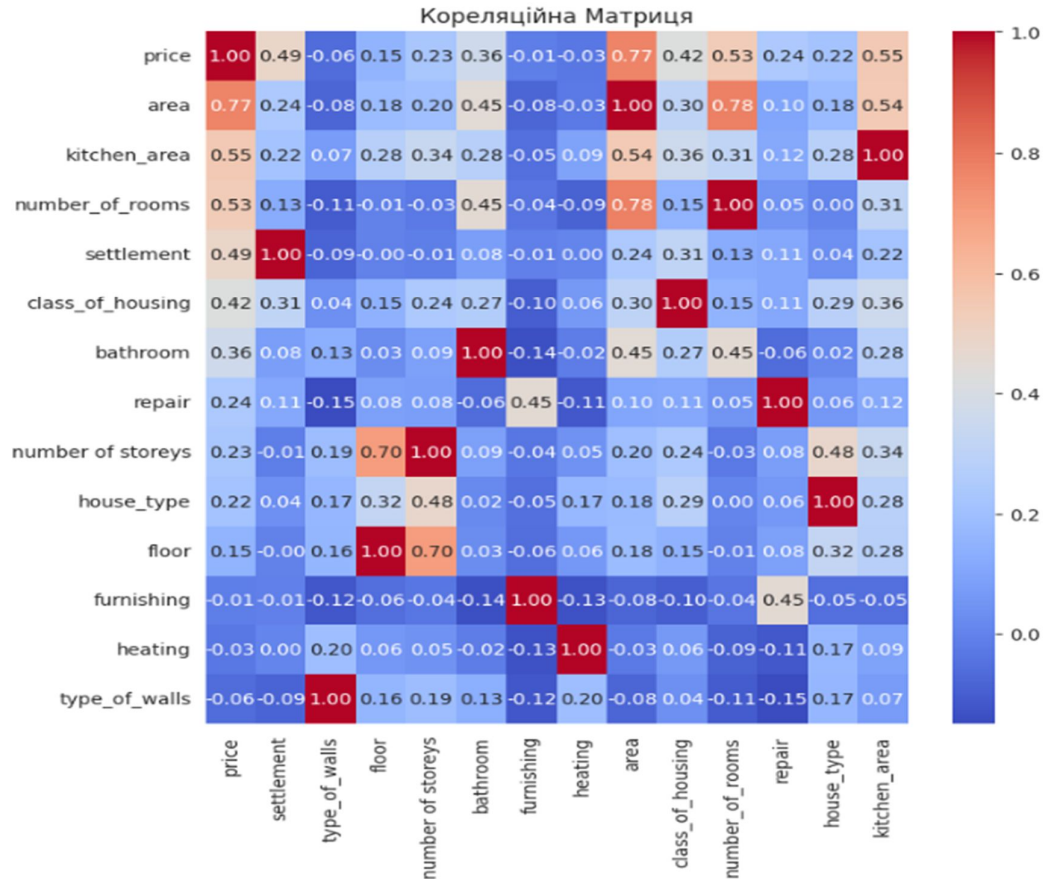


Рисунок 3.10 – Побудова кореляційної матриці

Джерело: розроблено автором на основі [69]

На основі побудованої кореляційної матриці, можна зробити наступні висновки, котрі потім буду застосовувати при побудові різних моделей, з урахуванням кореляції між ціною та іншими змінними.

Одним з найбільш помітних зв'язків є сильна позитивна кореляція між ціною та площею об'єкта нерухомості. Коефіцієнт кореляції 0,77 вказує на те, що зі збільшенням площі, як правило, зростає і ціна. Це цілком логічно, адже більша площа зазвичай розглядається як перевага на ринку нерухомості та підвищує привабливість об'єкта для потенційних покупців.

Крім того, спостерігається помірна позитивна кореляція між ціною та площею кухні з коефіцієнтом 0,55. Це може свідчити про те, що більші кухні також є бажаною характеристикою для покупців і можуть впливати на підвищення ціни.

Також район розташування об'єкта нерухомості, має досить високий позитивний коефіцієнт кореляції з ціною (0,49). Це означає що місце розташування об'єкта впливає на формування вартості об'єкту.

Помірна позитивна кореляція між ціною та класом житла з коефіцієнтом 0,42. Це може вказувати на те, що вищий клас житла (наприклад, елітне чи преміум-житло) зазвичай має вищу ціну порівняно з більш бюджетними варіантами типів житлових комплексів.

Існує позитивний зв'язок між ціною та кількістю кімнат з коефіцієнтом 0,53. Це узгоджується з логічним припущенням, що більша кількість кімнат робить об'єкт нерухомості більш привабливим для покупців і, відповідно, підвищує його вартість.

Аналізуючи кореляційну матрицю, можна помітити, що такі змінні, як опалення та наявність меблювання мають дуже слабкий або майже відсутній зв'язок з ціною нерухомості.

Зокрема, коефіцієнт кореляції між типом опалення в об'єкті нерухомості становить лише - 0,03, що свідчить про практично відсутню лінійну залежність між цими двома змінними. Це може означати, що тип системи опалення не є визначальним фактором для формування ціни на нерухомість у цьому наборі даних.

Схожа ситуація спостерігається і для змінної меблювання, яка має коефіцієнт кореляції з ціною на рівні -0,01. Такий низький показник кореляції вказує на те, що рівень оздоблення в об'єкті нерухомості, ймовірно, не відіграє суттєвої ролі у визначенні його вартості.

Нормалізація вхідних даних є необхідним етапом у процесі підготовки даних для моделювання та аналізу. Цей крок дозволяє приведення різних змінних до єдиного масштабу, що може значно покращити продуктивність та точність алгоритмів машинного навчання.

Оскільки мої вхідні дані містять змінні з різними діапазонами значень, це може призвести до домінування змінних з більшими значеннями над тими, що мають менші значення. Це може негативно вплинути на результати моделювання, оскільки алгоритми машинного навчання можуть надавати більшої ваги змінним з

більшими значеннями, ігноруючи вплив інших важливих змінних. Також, нормалізовані дані полегшують інтерпретацію результатів моделювання, оскільки всі змінні мають подібний масштаб. Це дозволяє легше порівнювати вагові коефіцієнти або важливість різних змінних у моделі.

Одним з популярних методів нормалізації даних є MinMaxScaler, який масштабує кожен змінну в діапазоні від 0 до 1. Цей метод застосовується до кожної змінної окремо, перетворюючи її значення за допомогою наступної формули [див. Формула (3.3)].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.3)$$

де  $x$  – оригінальне значення;

$x'$  – нормалізоване значення.

Після застосування MinMaxScaler, всі значення змінних в моєму наборі даних будуть знаходитися в діапазоні від 0 до 1. Це забезпечує, що жодна змінна не буде домінувати над іншими через свій більший діапазон значень (рис 3.11).

```
# Функція нормалізації даних
def normalize_dataframe(df):
    df_normalized = df.copy()

    numeric_cols = df_normalized.select_dtypes(include=['float64', 'int64']).columns.drop("price")
    scaler = MinMaxScaler(feature_range=(0, 1))
    df_normalized[numeric_cols] = scaler.fit_transform(df_normalized[numeric_cols])

    return df_normalized, scaler_price
```

Рисунок 3.11 – Програмна реалізація нормалізації даних за допомогою MinMaxScaler

*Джерело: розроблено автором на основі [70, 71]*

Отже, за результатами аналізу, та з урахування попередньої обробки усіх факторів що можуть впливати на об'єкт нерухомості, можна сформувати 4 вхідних датасети для побудови моделей:

1. Набір факторів котрий складається з усіх числових та категоріальних факторів представлених на матриці кореляції.

2. Набір факторів котрий складається лише з найбільш корельованих факторів (район розташування, тип та кількість санвузлів, площа об'єкта, клас будинку, кількість кімнат, площа кухні об'єкта нерухомості, тип будинку).
3. Набір факторів з першого датасету, з додавання текстових факторів котрі були попередньо оброблені з урахуванням одночасного кодування (тип комунікацій в об'єкті, тип побутової техніки, наявність мультимедійної техніки).
4. Датасет з урахуванням опису об'єкта нерухомості, що був оброблений за допомогою інструментарію NLP, та усіма ознаками з третього набору даних.

Для початку побудови моделей нейронних мереж, нам потрібно розділити дані на тестову та тренувальну вибірки. Це є необхідним кроком, оскільки дозволяє об'єктивно оцінити продуктивність моделі та уникнути проблеми переналаштування.

Тренувальна вибірка – це частина даних, яка використовується для навчання нейронної мережі. Під час процесу навчання, алгоритм аналізує ці дані, виявляє закономірності та налаштовує вагові коефіцієнти та параметри мережі таким чином, щоб мінімізувати функцію втрат.

Тестова вибірка – це решта даних, які не використовувалися для навчання моделі. Ця частина даних зберігається окремо та застосовується для оцінки продуктивності навченої нейронної мережі.

Вхідні дані було розділено у відповідності: 85% - становлять тренувальні дані, а 15% тестові.

Відправною точкою моделювання, буде застосування Ridge регресії для числових та категоріальних факторів котрі були перекодовані з застосування техніки кодування міток. Ridge регресія є одним з методів регуляризації лінійної регресії, який допомагає уникнути перенавчання моделі. Перенавчання виникає, коли модель занадто добре підігнана до навчальних даних і не може узагальнити нові дані належним чином. Ridge регресія вводить штраф на суму квадратів коефіцієнтів моделі, що змушує їх бути ближче до нуля, зменшуючи тим самим

вплив окремих ознак на модель. Це допомагає зробити модель більш стійкою до шуму в даних і покращує її узагальнюючу здатність (рис. 3.12) [72].

```
from sklearn.linear_model import Ridge
ridge_model = Ridge(alpha=0.5)
ridge_model.fit(X_train, y_train)
ridge_model_pred = ridge_model.predict(X_test)
```

Рисунок 3.12 – Застосування Ridge регресії

*Джерело: розроблено автором на основі [72, 73]*

За результатами моделювання, середньоквадратична похибка (RMSE) становить 1135804 грн. Це означає, що в середньому передбачені ціни на нерухомість відхиляються від фактичних значень приблизно на 1,1 млн. грн. Середня абсолютна похибка (MAE) дорівнює 908977 грн., що також є високим показником. Це означає, що в середньому абсолютні відхилення прогнозованих цін від реальних становлять близько 909 тис. грн. Така похибка є досить високою для оцінки вартості нерухомості або прийняття інвестиційних рішень, на основі цієї моделі [74, 75, 76].

Наступним етапом є перехід до моделювання з використанням нейронних мереж. Першою моделлю нейронної мережі, яку буде розроблено, буде найпростіша повнозв'язна мережа прямого поширення (рис. 3.13). Ця модель складається з одного прихованого шару та використовує лінійну функцію активації. Вона буде навчатися протягом 100 епох з використанням оптимізатора стохастичного градієнтного спуску та функції втрат середньоквадратичної помилки.

```
model = Sequential([
    Dense(13, input_shape=(13,), activation='linear')
    , Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=100)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
```

*Джерело: розроблено автором на основі [77]*

Рисунок 3.13 - Найпростіша повнозв'язна мережа прямого поширення з лінійною функцією активації

У цій простій моделі нейронної мережі перший прихований шар містить 13 нейронів, оскільки кількість нейронів у вхідному шарі дорівнює кількості вхідних змінних у наборі даних. Оскільки було відібрано 13 вхідних змінних, що описують різні характеристики нерухомості, встановлю таку ж кількість нейронів у першому прихованому шарі. Це дозволяє кожному нейрону обробляти інформацію від однієї вхідної змінної та виявляти відповідні закономірності.

Вихідний шар моделі містить лише один нейрон з лінійною функцією активації. Це пояснюється тим, що задача дипломної роботи полягає у прогнозуванні однієї числової цільової змінної – ціни нерухомості. Тому вихідний шар повинен мати лише один нейрон, який буде генерувати це єдине передбачене значення. Лінійна функція активації у вихідному шарі забезпечує безмежний діапазон можливих значень, що є бажаним для задачі регресії, оскільки я намагаюсь передбачити конкретне числове значення.

За результатами побудови простої моделі нейронної мережі з лінійною функцією активації, було отримано певні показники продуктивності на тестовій вибірці даних (табл. 3.4). Однак, перш ніж інтерпретувати ці метрики, слід зазначити, що передбачені значення ціни спочатку були нормалізовані. Тому для отримання реальних значень цін у тих самих одиницях вимірювання, що й вихідні дані, ми виконали зворотне перетворення, конвертувавши нормалізовані дані у звичайні.

Таблиця 3.4 – Показники метрик оцінювання простої нейронної мережі з лінійною функцією активації

| MAE         | RMSE        | MAPE   |
|-------------|-------------|--------|
| 1045262 грн | 1232664 грн | 21.9 % |

*Джерело: розроблено автором на основі [74-76]*

Загалом, результати простої моделі нейронної мережі з лінійною функцією активації для прогнозування цін на нерухомість виявилися не задовільними. Метрики продуктивності, свідчать про значні відхилення передбачених цін від фактичних значень.

Отримані результати можна пояснити кількома факторами. По-перше, лінійна модель може бути недостатньо потужною для моделювання складних нелінійних залежностей між характеристиками нерухомості та її ціною. По-друге, наявність лише одного шару обмежує здатність мережі до глибокого навчання та виявлення більш глибоких закономірностей у даних. Крім того, можливо, що кількість епох навчання була недостатньою для повної збіжності моделі.

Наступним кроком у процесі побудови моделей нейронних мереж для прогнозування цін на нерухомість буде створення моделі з використанням функції активації зрізаний лінійний вузол (рис. 3.14). Зрізаний лінійний вузол є однією з найпопулярніших нелінійних функцій активації, яка широко застосовується в сучасних архітектурах нейронних мереж.

Використання нелінійних функцій активації, таких як ReLU, може бути корисним з кількох причин. По-перше, вони дозволяють моделі ефективніше моделювати складні нелінійні залежності між вхідними даними та цільовою змінною. На відміну від лінійних функцій, нелінійні функції активації можуть виявляти більш складні взаємозв'язки та закономірності в даних, що є особливо важливим для задачі прогнозування цін на нерухомість.

По-друге, нелінійні функції активації допомагають уникнути проблеми зникаючого або вибухаючого градієнта, яка може виникати під час навчання глибоких нейронних мереж з лінійними активаціями. Ця проблема може призвести до повільної збіжності або навіть зупинки процесу навчання.

```
model = Sequential([
    Dense(13, input_shape=(13,), activation='relu')
    , Dense(1, activation='relu')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=100)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
```

*Джерело: розроблено автором на основі [51, 77]*

Рисунок 3.14 - Найпростіша повнозв'язна мережа прямого поширення з ReLU функцією активації

Після побудови моделі нейронної мережі з функцією активації ReLU, отримано наступні результати на тестовій вибірці (табл. 3.5):

Таблиця 3.5 – Показники метрик оцінювання простої нейронної мережі з ReLU функцією активації

| MAE         | RMSE        | MAPE   |
|-------------|-------------|--------|
| 1040611 грн | 1310601 грн | 19.9 % |

*Джерело: розроблено автором на основі [74-76]*

Порівнюючи ці метрики з результатами попередньої лінійної моделі, можна помітити деякі зміни, але загалом точність прогнозування залишається на низькому рівні.

RMSE (корінь із середнього квадрата помилки) збільшився з 1232664 до 1310601. Це означає, що в середньому передбачені значення ціни відхиляються від фактичних ще більше, ніж у лінійній моделі. Зростання RMSE свідчить про погіршення продуктивності моделі з функцією активації ReLU порівняно з лінійною.

MAE (середня абсолютна помилка) трохи зменшилася з 1045262 до 1040611. Це вказує на те, що середнє абсолютне відхилення між передбаченими та фактичними цінами зменшилося в моделі з ReLU.

MAPE (середня абсолютна відсоткова помилка), також трохи зменшилася, - з 21.9 % до 19.9 %.

Для знаходження оптимальної моделі нейронної мережі, здатної точно прогнозувати ціни на нерухомість, було вирішено збільшити кількість прихованих шарів, оскільки більш глибокі архітектури можуть краще виявляти складні нелінійні залежності у даних (рис. 3.15).

Також, оскільки, попередні моделі з одним шаром, не мали достатньої потужності для ефективного моделювання взаємозв'язків між характеристиками нерухомості та її ціною, додавання додаткових прихованих шарів дозволяє мережі вивчати більш складні ознаки, що може покращити її здатність до узагальнення та підвищити точність прогнозування.

Крім того, було вирішено спробувати використати різні функції активації для прихованих шарів, оскільки вибір функції активації може суттєво вплинути на ефективність моделі. Різні функції активації мають різні властивості та можуть

краще підходити для моделювання певних типів залежностей. Наприклад, функція ReLU добре справляється з нелінійними залежностями.

```
model = Sequential([
    Dense(128, input_shape=(13,), activation='linear')
    , Dense(64, activation='relu')
    , Dense(32, activation='relu')
    , Dense(16, activation='relu')
    , Dense(8, activation='relu')
    , Dense(4, activation='relu')
    , Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=100)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
```

*Джерело: розроблено автором на основі [51, 77]*

Рисунок 3.15 - Повнозв'язна мережа прямого поширення з 5 прихованими шарами

У цій моделі ми маємо п'ять прихованих шарів з різною кількістю нейронів. Перший шар використовує лінійну функцію активації, тоді як наступні п'ять приховані вузли використовують функцію ReLU. Вихідний шар також має лінійну активацію, оскільки ми намагаємося передбачити числове значення ціни. Модель компілюється з використанням оптимізатора стохастичного градієнтного спуску (SGD) та функції втрат середньоквадратичної помилки (MSE), як і в попередніх моделях.

Після побудови моделі нейронної мережі з п'ятьма прихованими шарами та різними функціями активації, отримано наступні результати на тестовій вибірці (табл. 3.6):

Таблиця 3.6 – Показники метрик оцінювання нейронної мережі з 5 прихованими вузлами

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 936228 грн | 1106431 грн | 19.3 % |

*Джерело: розроблено автором на основі [74-76]*

Порівнюючи ці метрики з попередніми моделями, можна помітити покращення точності прогнозування. Збільшення кількості прихованих шарів, та використання різних функцій активації, ймовірно, допомогло моделі краще

виявляти складні нелінійні залежності між характеристиками нерухомості та її ціною.

Середня абсолютна помилка (MAE) зменшилася з 1040611 грн у моделі з ReLU функцією активації до 936228 грн у поточній моделі. Це означає, що в середньому передбачені ціни відхиляються від фактичних на 936 тисяч гривень.

Корінь із середнього квадрата помилки також зменшився з 1310601 грн у попередній моделі, до 1106431 грн у поточній. Це свідчить про те, що великі відхилення між передбаченими та фактичними цінами стали менш поширеними, що є позитивним результатом.

Середня абсолютна відсоткова помилка (MAPE) становить 19.3%, що також є покращенням порівняно з 19.9 % у попередній моделі. Це означає, що в середньому передбачені ціни відхиляються від фактичних приблизно на 19.3%.

Загалом, ці результати свідчать про те, що збільшення кількості прихованих шарів та використання різних функцій активації допомогло моделі краще вивчити складні закономірності у даних та підвищити точність прогнозування цін на нерухомість. Глибша архітектура мережі, ймовірно, дозволила виявити більш абстрактні та складні ознаки, що є важливими для прогнозування ціни.

Наступним кроком для подальшого вдосконалення моделі нейронної мережі буде збільшення кількості епох навчання до 200. Це може бути корисним, оскільки більша кількість епох дозволяє моделі продовжувати навчатися та краще вивчати складні закономірності у даних (рис. 3.16).

У попередній моделі ми використовували лише 100 епох, що може бути недостатнім для повної збіжності моделі та досягнення оптимального рівня продуктивності. Збільшення кількості епох до 200 може допомогти моделі краще узагальнити дані та знайти більш точні вагові коефіцієнти для кожного нейрона.

```

model = Sequential([
    Dense(128, input_shape=(13,), activation='linear')
    , Dense(64, activation='relu')
    , Dense(32, activation='relu')
    , Dense(16, activation='relu')
    , Dense(8, activation='relu')
    , Dense(4, activation='relu')
    , Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=200)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)

```

*Джерело: розроблено автором на основі [51, 77]*

Рисунок 3.16 - Повнозв'язна мережа прямого поширення з 5 прихованими вузлами та збільшеною кількістю епох навчання

У цій моделі було збережено архітектуру з п'ятьма прихованими шарами та різними функціями активації, як у попередній моделі. Однак, кількість епох навчання збільшено до 200. Модель компілюється з використанням оптимізатора стохастичного градієнтного спуску (SGD) та функції втрат середньоквадратичної помилки (MSE), як і раніше.

Після побудови моделі нейронної мережі з п'ятьма прихованими шарами, різними функціями активації та збільшенням кількості епох навчання до 200, було отримано наступні результати на тестовій вибірці (табл. 3.7):

Таблиця 3.7 – Показники метрик оцінювання нейронної мережі з 5 прихованими вузлами та збільшеною кількістю епох навчання

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 900904 грн | 1087584 грн | 18.5 % |

*Джерело: розроблено автором на основі [74-76]*

Середня абсолютна помилка дещо покращилася, зменшившись з 936228 грн до 900904 грн. Це означає, що в середньому передбачені ціни відхиляються від фактичних на трохи меншу суму, що є позитивним результатом. Середня абсолютна відсоткова помилка також дещо зменшилася, з 19.3% до 18.5%. Це означає, що в середньому передбачені ціни відхиляються від фактичних на трохи менший відсоток, що є позитивним результатом.

Для подальшого вдосконалення моделі нейронної мережі та потенційного покращення точності прогнозування цін на нерухомість, ми можемо спробувати використати функцію активації Leaky ReLU (Rectified Linear Unit) замість звичайної ReLU (рис. 3.17). Функція Leaky ReLU може бути корисною з кількох причин.

По-перше, одна з основних проблем звичайної функції ReLU полягає в тому, що вона може призвести до "мертвих" нейронів, коли вхідні значення стають негативними. Це означає, що градієнт для цих нейронів стає нулем, і вони перестають навчатися та оновлювати свої вагові коефіцієнти. Функція Leaky ReLU вирішує цю проблему, дозволяючи невеликим негативним значенням проходити через нейрон з невеликим нахилом, замість того, щоб повністю їх відкидати.

По-друге, використання Leaky ReLU може допомогти боротися з проблемою зникаючого градієнта, яка часто виникає під час навчання глибоких нейронних мереж. Завдяки тому, що негативні значення не повністю відкидаються, градієнт може продовжувати поширюватися через шари мережі, що полегшує процес навчання.

```
model = Sequential([
    Dense(128, input_shape=(13,)), activation='linear')
    , Dense(64)
    , LeakyReLU(alpha=0.01)
    , Dense(32)
    , LeakyReLU(alpha=0.01)
    , Dense(16)
    , LeakyReLU(alpha=0.01)
    , Dense(8)
    , LeakyReLU(alpha=0.01)
    , Dense(4)
    , LeakyReLU(alpha=0.01)
    , Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
history = model.fit(X_train, y_train, epochs=200)
evaluate(model, X_test, y_test, scaler.price, normalized=True)
```

*Джерело: розроблено автором на основі [51, 77]*

Рисунок 3.17 - Повнозв'язна мережа прямого поширення з 5 прихованими шарами та LeakyReLU функцією активації в прихованих шарах

Після побудови моделі нейронної мережі з використанням функції активації Leaky ReLU у прихованих шарах, ми отримали наступні результати на тестовій вибірці (табл. 3.8):

Таблиця 3.8 – Показники метрик оцінювання нейронної мережі з 5 прихованими вузлами та LeakyReLU функцією активації

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 876670 грн | 1072787 грн | 17.7 % |

*Джерело: розроблено автором на основі [74-76]*

Порівнюючи ці метрики з попередньою моделлю, яка використовувала звичайну функцію активації ReLU в прихованих шарах, можна помітити деякі зміни в продуктивності.

Середня абсолютна помилка зменшилася з 900904 грн до 876670 грн. Це означає, що в середньому передбачені ціни відхиляються від фактичних на меншу суму, що є позитивним результатом.

Корінь із середнього квадрата помилки (RMSE) також зменшився з 1087584 грн до 1072787 грн. Це може свідчити про те, що модель з Leaky ReLU стала менш чутливою до великих відхилень між передбаченими та фактичними цінами, що призвело до зменшення RMSE.

Середня абсолютна відсоткова помилка (MAPE) значно зменшилася з 18.5% до 17.7%. Це означає, що в середньому передбачені ціни відхиляються від фактичних на менший відсоток, що є дуже позитивним результатом.

Загалом, ці результати свідчать про те, що використання функції активації Leaky ReLU в прихованих шарах призвело до покращення продуктивності моделі порівняно з попередньою версією з ReLU. Зменшення MAE, RMSE та MAPE вказує на те, що модель з Leaky ReLU стала більш точною у прогнозуванні цін на нерухомість, зменшивши як середні, так і великі відхилення між передбаченими та фактичними значеннями.

Після побудови та оцінки кількох моделей нейронних мереж для прогнозування цін на нерухомість, було прийнято рішення про пошук оптимальної кількості епох навчання. Кількість епох є гіперпараметром, який визначає, скільки

разів модель буде проходити через весь набір тренувальних даних під час процесу навчання. Занадто мала кількість епох може призвести до недонавчання моделі, коли вона не встигає повністю вивчити закономірності у даних. Водночас, занадто велика кількість епох може спричинити переналаштування, коли модель "запам'ятовує" специфічні деталі тренувальних даних, включаючи шум та випадкові коливання, замість того, щоб узагальнювати загальні закономірності.

Для визначення оптимальної кількості епох було вирішено використати візуалізацію середньої абсолютної помилки (MAE) на тестових даних.

Візуалізація MAE на тестових даних під час процесу навчання дозволяє відстежувати, як змінюється продуктивність моделі з кожною епохою.

Мінімальне значення MAE на тестових даних відповідає оптимальній кількості епох, коли модель досягає найкращої продуктивності та здатності до узагальнення. Подальше збільшення кількості епох після цієї точки призводить до переналаштування моделі та зростання помилки на тестових даних.

Для того, щоб протестувати різні підходи до оптимізації та навчання моделей нейронних мереж, було вирішено виконати їх з використанням різних оптимізаторів.

Оскільки раніше вже було використано стохастичний градієнтний спуск як оптимізатор для навчання моделей нейронних мереж спробую знайти оптимальну кількість епох для цього оптимізатора використовуючи модель з 5 прихованими шарами та LeakyReLU функцією активації як і в попередній моделі (рис. 3.18).

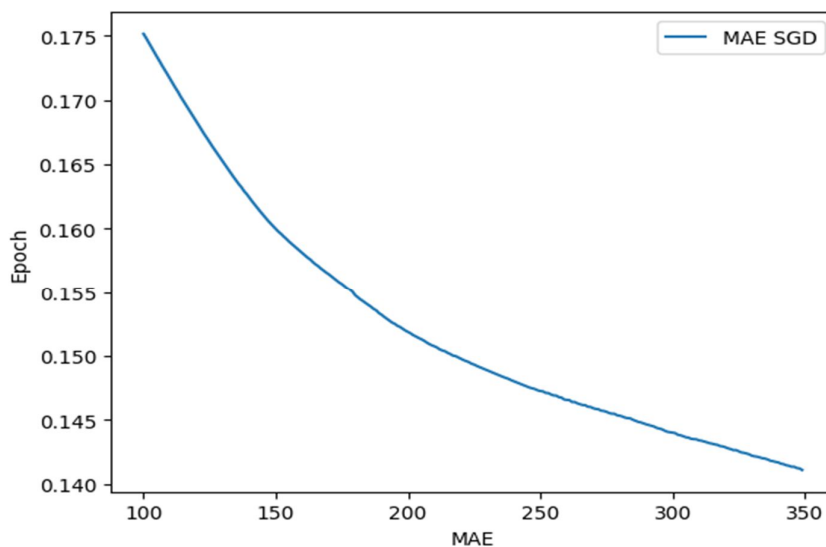


Рисунок 3.18 – Візуалізація середньої абсолютної похибки на тренувальних даних з оптимізатором стохастичний градієнтний спуск

*Джерело: розроблено автором на основі [78]*

Отже, за результатами візуалізації, можна побачити що збільшення кількості епох до 350 призводить до зменшення середньої абсолютної помилки на тренувальних даних. Результати метрик оцінювання на тестовій вибірці мають наступні показники (табл. 3.9):

Таблиця 3.9 – Показники метрик оцінювання нейронної мережі з 5 прихованими вузлами, LeakyReLU функцією активації, 350 епох навчання та SGD оптимізатором

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 822620 грн | 1057200 грн | 16.9 % |

*Джерело: розроблено автором на основі [74-76]*

MAE становить 822620 грн, що означає, що в середньому передбачені ціни відхиляються від фактичних на суму близько 822 тисячі гривень. Це є досить прийнятним рівнем помилки порівнюючи з попередніми моделями.

RMSE дорівнює 1057200 грн. Значення RMSE вказує на те, що хоча середня абсолютна помилка є відносно невеликою, існують окремі випадки, коли передбачені ціни значно відрізняються від фактичних.

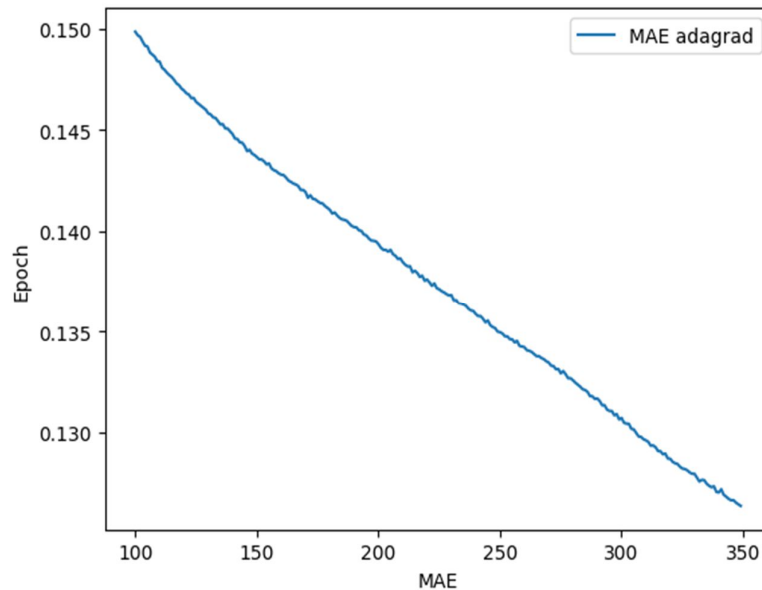
Порівнюючи ці метрики з попередніми результатами, можна помітити, що збільшення кількості епох навчання дозволило зменшити значення MAE, RMSE та MAPE. Це свідчить про те, що додаткове навчання моделі протягом більшої кількості епох допомогло їй краще вивчити закономірності у даних та покращити точність прогнозування цін на нерухомість.

Наступним кроком знаходження оптимальної кількості епох для моделі нейронної мережі, буде спроба знайти оптимальну кількість епох з урахування оптимізатора Adagrad.

Adagrad є вдосконаленою версією SGD, яка автоматично адаптує коефіцієнт навчання для кожного вагового коефіцієнта на основі історії градієнтів. Це дозволяє швидше навчати ваги з невеликими градієнтами та повільніше навчати ваги з великими градієнтами. Adagrad може забезпечити кращу збіжність порівняно з

SGD, але може страждати від зменшення коефіцієнта навчання до занадто малих значень після багатьох ітерацій.

Для побудови моделі з оптимізатором Adagrad та знаходженням оптимальної кількості епох навчання, я залишу модель з LeakyReLU функцією активації та 5 прихованими шарами (рис. 3.19).



*Джерело: розроблено автором на основі [78]*

Рисунок 3.19 – Візуалізація середньої абсолютної похибки на тренувальних даних з оптимізатором Adagrad

Отже, за результатами графіку, видно що збільшення кількості епох до призводить до зменшення середньої абсолютної помилки на тренувальних даних.

Аналізуючи результати з використання оптимізатора Adagrad на тренувальній вибірці (табл. 3.10), можна помітити, що метрики демонструють гірші показники порівняно з попередніми експериментами з оптимізатором SGD.

Таблиця 3.10 – Показники метрик оцінювання нейронної мережі з 5 прихованими вузлами, LeakyReLU функцією активації, 350 епох навчання та Adagrad оптимізатором

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 852831 грн | 1092481 грн | 17.3 % |

*Джерело: розроблено автором на основі [74-76]*

Незважаючи на те, що середня абсолютна помилка (MAE) на тренувальних даних була меншою під час навчання з Adagrad, це не призвело до покращення продуктивності моделі на тестових даних. Навпаки, всі три метрики помилки (RMSE, MAE та MAPE) зросли, що свідчить про зниження точності прогнозування цін на нерухомість.

### **3.3 Аналіз результатів предиктивного моделювання вартості житла на базі алгоритмів оптимізації гіперпараметрів та інструментарію автоматизованого машинного навчання**

У попередніх моделях, які були побудовані, використовувались як числові, так і категоріальні фактори котрі були приведені за допомогою Label Encoding, що дозволило перетворити їх на числові значення, придатні для використання в моделях машинного навчання.

Для поліпшення якості моделі та отримання більш точних прогнозів, у далі буде побудована модель, яка враховуватиме лише найбільш корельовані фактори.

Для цього залишу в якості факторів наступні змінні: район розташування, тип ванної кімнати, площа об'єкту, клас будинку, тип будинку, кількість кімнат, площа кухні.

Оскільки пошук оптимальних гіперпараметрів раніше відбувався вручну, було вирішено автоматизувати цей процес.

Для автоматизації цього процесу використовується бібліотека KerasTuner, яка дозволяє проводити автоматичний пошук найкращих гіперпараметрів для моделей, побудованих на основі Keras.

Одним з методів пошуку, який реалізований у KerasTuner, є RandomSearch. Цей метод випадковим чином генерує набори гіперпараметрів у заданому діапазоні значень та оцінює продуктивність моделі з цими гіперпараметрами (рис. 3.20).

```

# Функція для створення моделі
def build_model(hp):
    model = keras.Sequential()

    # Визначення гіперпараметрів для тюнінга
    hp_units_1 = hp.Int('units_1', min_value=32, max_value=256, step=32)
    hp_units_2 = hp.Int('units_2', min_value=16, max_value=128, step=16)
    hp_units_3 = hp.Int('units_3', min_value=8, max_value=64, step=8)
    hp_units_4 = hp.Int('units_4', min_value=4, max_value=32, step=4)
    hp_units_5 = hp.Int('units_5', min_value=2, max_value=16, step=2)

    # Додавання слоїв в модель
    model.add(layers.Dense(units=hp_units_1, activation='linear', input_shape=(7,)))
    model.add(layers.Dense(units=hp_units_2))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(units=hp_units_3))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(units=hp_units_4))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(units=hp_units_5))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(1, activation='linear'))
    model.compile(optimizer='sgd', loss='mse')
    return model

tuner = RandomSearch(
    build_model,
    objective='val_loss',
    max_trials=8,
    executions_per_trial=2,
    directory='project_yaroslav',
    project_name='diplom'
)
for train_index, val_index in kfold.split(X_train):
    tuner.search(X_train[train_index], y_train[train_index],
                epochs=200,
                validation_data=(X_train[val_index], y_train[val_index]))

# Отримання найкращої моделі
best_model = tuner.get_best_models(num_models=1)[0]

# Вивід інформації про архітектуру мережі
best_model.summary()
evaluate_model(best_model, X_test, y_test, scaler_price, normalized= True)

```

Рисунок 3.20 – Автоматизація вибору оптимальних гіперпараметрів за допомогою RandomSearch

*Джерело: розроблено автором на основі [51, 79]*

Для реалізації RandomSearch, спочатку визначається функція, яка створює послідовну модель нейронної мережі з декількома прихованими шарами та налаштовуваними гіперпараметрами, такими як кількість нейронів у кожному шарі. Потім створюється об'єкт, який ініціалізується з функцією build\_model, цільовою функцією втрат, максимальною кількістю ітерацій пошуку та іншими параметрами. Далі відбувається цикл крос-валідації, де на кожній ітерації викликається метод tuner.search. Під час пошуку, RandomSearch генерує різні комбінації гіперпараметрів, створює моделі з цими гіперпараметрами, тренує їх на

навчальному наборі даних та оцінює їх продуктивність на валідаційному наборі даних. Після завершення пошуку, отримується найкраща модель.

Згідно результатів моделювання, найкраща архітектура нейронної мережі, для найбільш важливих факторів, представлена на (рис. 3.21).

Model: "sequential"

| Layer (type)              | Output Shape | Param # |
|---------------------------|--------------|---------|
| dense (Dense)             | (None, 256)  | 2,048   |
| dense_1 (Dense)           | (None, 48)   | 12,336  |
| leaky_re_lu (LeakyReLU)   | (None, 48)   | 0       |
| dense_2 (Dense)           | (None, 64)   | 3,136   |
| leaky_re_lu_1 (LeakyReLU) | (None, 64)   | 0       |
| dense_3 (Dense)           | (None, 12)   | 780     |
| leaky_re_lu_2 (LeakyReLU) | (None, 12)   | 0       |
| dense_4 (Dense)           | (None, 6)    | 78      |
| leaky_re_lu_3 (LeakyReLU) | (None, 6)    | 0       |
| dense_5 (Dense)           | (None, 1)    | 7       |

Рисунок 3.21 – Архітектура нейронної мережі побудованою за допомогою RandomSearch алгоритму

*Джерело: розроблено автором на основі [51, 79]*

За результати побудованих моделей, найкраща з них показала наступні результати на тестовій вибірці (табл. 3.11).

Таблиця 3.11 – Показники метрик оцінювання нейронної мережі побудованої за допомогою RandomSearch алгоритму

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 927903 грн | 1168596 грн | 18.4 % |

*Джерело: розроблено автором на основі [74-76]*

Незважаючи на те, що в поточній моделі була використана менша кількість факторів, зменшення кількості факторів не призвело до покращення продуктивності моделі. Навпаки, результати трохи погіршилися порівняно з попередньою моделлю, яка використовувала більше факторів. Це може бути

пов'язано з тим, що видалені фактори могли містити важливу інформацію для прогнозування, і їх відсутність негативно вплинула на здатність моделі точно передбачати цільову змінну.

Наступним кроком дипломної роботи буде побудова моделі з використанням Hyperband нейронної мережі. Hyperband є ефективним алгоритмом для налаштування гіперпараметрів, який дозволяє значно скоротити час та обчислювальні ресурси, необхідні для пошуку оптимальної конфігурації гіперпараметрів.

Перевагою застосування Hyperband є його ефективність: він значно ефективніший, ніж випадковий пошук, особливо коли простір гіперпараметрів є великим. Він дозволяє швидко відкидати неперспективні конфігурації та зосереджуватися на більш перспективних областях простору гіперпараметрів. Завдяки адаптивному розподілу ресурсів, Hyperband може значно скоротити час та обчислювальні ресурси, необхідні для пошуку оптимальної конфігурації гіперпараметрів.

Застосовувати Hyperband я буду для датасету котрий складається з усіх числових та текстових факторів оброблених за Label Encoding, та з текстових факторів (тип комунікацій в об'єкті, тип побутової техніки, наявність мультимедійної техніки) котрі були оброблені One Hot Encoding.

За результатами побудови мережі з використанням Hyperband алгоритму, було змодельовано наступну мережу (рис 3.22).

Model: "sequential"

| Layer (type)              | Output Shape | Param # |
|---------------------------|--------------|---------|
| dense (Dense)             | (None, 128)  | 8,064   |
| dense_1 (Dense)           | (None, 32)   | 4,128   |
| leaky_re_lu (LeakyReLU)   | (None, 32)   | 0       |
| dense_2 (Dense)           | (None, 24)   | 792     |
| leaky_re_lu_1 (LeakyReLU) | (None, 24)   | 0       |
| dense_3 (Dense)           | (None, 8)    | 200     |
| leaky_re_lu_2 (LeakyReLU) | (None, 8)    | 0       |
| dense_4 (Dense)           | (None, 3)    | 27      |
| leaky_re_lu_3 (LeakyReLU) | (None, 3)    | 0       |
| dense_5 (Dense)           | (None, 1)    | 4       |

Рисунок 3.22 – Архітектура нейронної мережі побудованою за допомогою Hyperband алгоритму

*Джерело: розроблено автором на основі [51, 79]*

За результатами оцінювання метрик (табл. 3.12) додавання додаткових факторів не призвело до суттєвої зміни характеристик основних метрик оцінювання моделі на тестових даних. Це може свідчити про те, що додаткові фактори не вносять значної нової інформації для покращення прогнозування цільової змінної.

Таблиця 3.12 – Показники метрик оцінювання нейронної мережі побудованої за допомогою Hyperband

| MAE        | RMSE        | MAPE |
|------------|-------------|------|
| 964979 грн | 1215340 грн | 19 % |

*Джерело: розроблено автором на основі [74-76]*

У наступному етапі дослідження буде використовуватися новий набір даних, який об'єднує попередній набір даних з додатковою інформацією, отриманою за допомогою інструментарію NLP. Цей новий датасет включатиме всі наявні ознаки з минулого набору даних, а також додаткові ознаки, отримані шляхом обробки текстових описів об'єктів нерухомості.

Об'єднання цих нових ознак, отриманих з описів об'єктів нерухомості, з існуючими ознаками з минулого набору даних може забезпечити більш повне та інформативне представлення даних.

Окрім використання нового набору даних, який включає ознаки, отримані за допомогою інструментарію NLP, в цьому дослідженні також буде застосовано метод Байєсівської оптимізації (Bayesian Optimization) для налаштування гіперпараметрів моделі машинного навчання.

Байєсівська оптимізація є ефективним підходом для пошуку оптимальних значень гіперпараметрів, які впливають на продуктивність моделі. Цей метод поєднує в собі статистичні методи та машинне навчання для інтелектуального пошуку в просторі гіперпараметрів, замість випадкового або вичерпного перебору.

Основною перевагою Байєсівської оптимізації є її здатність ефективно знаходити оптимальні рішення з меншою кількістю ітерацій порівняно з іншими методами оптимізації.

На відміну від традиційних методів оптимізації, таких як випадковий пошук або вичерпний перебір, Байєсівська оптимізація використовує інформацію про попередні ітерації для покращення наступних кроків пошуку.

Загалом, застосування Байєсівської оптимізації в поєднанні з новим набором даних, збагаченим текстовим описом об'єкта нерухомості, може забезпечити більш точну модель для прогнозування ціни на нерухомість.

За результатами побудови нейронної мережі з Байєсівської оптимізацією було отримано наступну найбільш оптимальну модель (рис. 3.23)

| Layer (type)              | Output Shape | Param # |
|---------------------------|--------------|---------|
| dense (Dense)             | (None, 32)   | 800     |
| dense_1 (Dense)           | (None, 112)  | 3,696   |
| leaky_re_lu (LeakyReLU)   | (None, 112)  | 0       |
| dense_2 (Dense)           | (None, 32)   | 3,616   |
| leaky_re_lu_1 (LeakyReLU) | (None, 32)   | 0       |
| dense_3 (Dense)           | (None, 12)   | 396     |
| leaky_re_lu_2 (LeakyReLU) | (None, 12)   | 0       |
| dense_4 (Dense)           | (None, 8)    | 104     |
| leaky_re_lu_3 (LeakyReLU) | (None, 8)    | 0       |
| dense_5 (Dense)           | (None, 1)    | 9       |

Рисунок 3.23 - Архітектура нейронної мережі побудованою за допомогою Байєсівської оптимізації

*Джерело: розроблено автором на основі [51, 79]*

За результатами побудованої моделі (табл. 3.13), котра використовувала Байєсівську оптимізацію, а також фактор текстового опису об'єкта нерухомості, можна зробити висновок що ця модель показала досить гарні результати, виходячі з основних метрик якості моделі.

Додавання нової ознаки, призвело до покращення результатів, та зменшення RMSE та MAE у порівнянні з моделлю котра використовувала лише найбільш корельовані фактори впливу на об'єкт нерухомості, та моделлю в якій було додано текстові фактори перекодовані за допомогою Label Encoding.

Таблиця 3.13 – Показники метрик оцінювання нейронної мережі побудованої за допомогою Байєсівської оптимізації

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 881508 грн | 1054657 грн | 17.4 % |

*Джерело: розроблено автором на основі [74-76]*

Заключним етапом роботи, є використання автоматизованих інструментів побудови моделей нейронних мереж. Оскільки процес налаштування архітектури нейронної мережі є складним процесом, котрий вимагає багато ітерацій та

експериментів, використання автоматизованих підходів, таких як AutoML та AutoKeras, для створення моделей глибокого навчання, є оптимальним рішенням.

AutoKeras має перевагу у порівнянні з ручним налаштуванням, оскільки автоматизовані інструменти використовують передові алгоритми та методи для пошуку оптимальної архітектури моделі та підбору гіперпараметрів, що призводить до кращої продуктивності моделей у порівнянні з ручним налаштуванням [80].

Результати, отримані за допомогою AutoKeras (табл. 3.14), демонструють, що побудована модель (рис.3.24) є найкращою порівняно з усіма моделями, які були розроблені вручну. Це підтверджується показниками якості моделі, такими як корінь RMSE, середня абсолютна помилка (MAE) та середня абсолютна відсоткова помилка (MAPE).

| Layer (type)                    | Output Shape | Param # |
|---------------------------------|--------------|---------|
| input_layer (InputLayer)        | (None, 24)   | 0       |
| cast (Cast)                     | (None, 24)   | 0       |
| cast_to_float32 (CastToFloat32) | (None, 24)   | 0       |
| dense (Dense)                   | (None, 512)  | 12,800  |
| re_lu (ReLU)                    | (None, 512)  | 0       |
| dense_1 (Dense)                 | (None, 512)  | 262,656 |
| re_lu_1 (ReLU)                  | (None, 512)  | 0       |
| dense_2 (Dense)                 | (None, 128)  | 65,664  |
| re_lu_2 (ReLU)                  | (None, 128)  | 0       |
| dropout (Dropout)               | (None, 128)  | 0       |
| regression_head_1 (Dense)       | (None, 1)    | 129     |

Рисунок 3.24 - Архітектура нейронної мережі побудованою за допомогою автоматизованого машинного навчання

*Джерело: розроблено автором на основі [80]*

Модель, побудована за допомогою AutoKeras, продемонструвала найнижче значення MAE, а саме 781489 грн, порівняно з усіма іншими моделями, що свідчить про її високу точність у прогнозуванні цін на нерухомість.

Також, за результатами моделювання слід зробити висновок про те, що в середньому змодельовані значення відхиляються від початкових на 16.5%.

Таблиця 3.14 – Показники метрик оцінювання нейронної мережі побудованої за допомогою автоматизованого машинного навчання

| MAE        | RMSE        | MAPE   |
|------------|-------------|--------|
| 781489 грн | 1004340 грн | 16.5 % |

*Джерело: розроблено автором на основі [74-76]*

## ВИСНОВКИ

У магістерській роботі було досліджено застосування Data Driven підходу та передових методів машинного і глибокого навчання для вирішення актуальної задачі оптимального вибору об'єктів нерухомості для споживачів та інвесторів.

Метою роботи було розробити ефективні моделі для визначення найбільш привабливих житлових об'єктів з урахуванням різноманітних факторів, що дозволить зацікавленим сторонам обґрунтовано обирати найкращі варіанти для придбання або інвестування.

На початку роботи було проведено ґрунтовний аналіз сучасного стану ринку нерухомості України, виявлено ключові тенденції, фактори впливу та регіональні особливості.

У роботі проведено ґрунтовне дослідження теоретичних основ Data-Driven підходу, його витоків, фундаментальних принципів та ключових етапів реалізації. Визначено, що Data-Driven підхід є потужною методологією прийняття рішень, яка ґрунтується на аналізі та інтерпретації емпіричних даних. Розглянуто зв'язок Data Driven підходу з теорією прийняття рішень та його роль у сучасній методології прийняття рішень на основі даних. Особливу увагу приділено обґрунтуванню важливості Data-Driven підходу саме у сфері нерухомості.

Ретельно досліджено існуючі методи та підходи у сфері прогнозування цін на нерухомість. Було проаналізовано переваги та недоліки таких популярних методів машинного навчання, як лінійна регресія, дерева рішень, випадкові ліси, градієнтний бустинг та інші. Виявлено, що ці методи можуть ефективно виявляти приховані закономірності та взаємозв'язки у великих наборах даних, що містять безліч факторів, які впливають на ціни на нерухомість. Виявлено їх обмеження у моделюванні складних нелінійних залежностей, що зумовило необхідність застосування потужніших методів глибокого навчання.

Досліджено теоретичні засади нейронних мереж, їх архітектури, функції активації, алгоритми навчання. Проаналізовано особливості застосування

нейронних мереж для задач оцінки та вибору об'єктів нерухомості. Окремо розглянуто методи автоматизованого машинного навчання для автоматичного пошуку оптимальних архітектур та налаштування гіперпараметрів.

Важливим етапом роботи стала підготовка та обробка вхідних даних, необхідних для аналізу ціни на об'єкти нерухомості. Було зібрано та оброблено дані про тисячі об'єктів нерухомості, включаючи числові, категоріальні та текстові дані. Розроблено комплексний підхід до обробки різних типів даних, який включав техніки кодування, нормалізації, лематизації, векторизації тексту та інші. Застосовано спеціалізовані методи обробки природної мови (NLP) для аналізу текстових описів об'єктів нерухомості та перетворення цієї інформації у числовий формат, придатний для використання в моделях машинного навчання.

На основі підготовлених даних було побудовано та протестовано численні моделі нейронних мереж з різною архітектурою, функціями активації, оптимізаторами та гіперпараметрами. Проведено ретельний аналіз продуктивності розроблених моделей за допомогою основних метрик оцінювання

За результатами моделювання виявлено, що додавання текстових описів об'єктів нерухомості, оброблених за допомогою NLP, може суттєво покращити точність прогнозування цін. Це пояснюється тим, що текстові описи містять важливу додаткову інформацію про об'єкти, яка не може бути повністю відображена у числових та категоріальних факторах.

Також, результати роботи демонструють потужність автоматизованих інструментів машинного навчання для створення ефективних моделей глибокого навчання, що забезпечують вищу продуктивність порівняно з ручним налаштуванням.

За результатми проробленої роботи, досягнуто результату, за якого середня абсолютна похибка на тренувальній вибірці становить 16,5%.

Загалом, представлена робота демонструє потенціал DataDriven підходу та методів машинного і глибокого навчання для вирішення складних задач аналізу та оцінки об'єктів нерухомості. Отримані результати відкривають нові можливості для підвищення ефективності прийняття рішень на ринку нерухомості.

Перспективами подальших досліджень можуть бути розробка складніших архітектур нейронних мереж, інтеграція додаткових джерел даних, вдосконалення методів автоматизованого машинного навчання, а також адаптація розроблених підходів до інших сфер, де потрібно здійснювати оцінку та вибір на основі великих даних.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Міщенко В. І., Неклюдова Т. М. Теоретичні засади циклічності економічного розвитку та необхідність проциклічного регулювання економіки // Економічні студії. — 2018. — № 3 (21). — С. 22–27.
2. Стан ринку нерухомості України 2024: ризики і можливості повоєнного відновлення міст. URL: <https://blog.liga.net/user/sshyinkarenko/article/53600>. (дата звернення 15.10.2024).
3. Оцінка та управління нерухомістю: навчальний посібник / [В. Р. Кучеренко, М. А. Заєць, О. В. Захарченко, Н. В. Сментина, В. О. Улибіна]. — Одеса: Видавництво ТОВ «Лерадрук», 2013. — 272 с.
4. Шибріна С. Ринок Нерухомого майна як економічна категорія та об'єкт статистичного дослідження/ С. Шибріна // Вісник Київського національного університету ім.. Т. Шевченка. — 2011. Том 1. — С. 58-62.
5. Запотоцька В. А. Ціноутворення на регіональних ринках житла, теоретичний та прикладний аспект/ В. А. Запотоцька // Вісник Київського національного університету ім.. Т. Шевченка. — 2017. Том 1. — С. 138-146.
6. Глобальний прогноз на 2024 рік. Нові тенденції в сфері нерухомості. URL: [https://propertytimes.com.ua/trends/globalniy\\_prognoz\\_na\\_2024\\_rik\\_novi\\_tendentsiyi\\_v\\_sferi\\_neruhomosti](https://propertytimes.com.ua/trends/globalniy_prognoz_na_2024_rik_novi_tendentsiyi_v_sferi_neruhomosti). (дата звернення: 17.10.2024).
7. Вплив війни на ринок нерухомості в Україні: прогноз на 2025 рік URL: <https://riel.ua/blogs/vpliv-viini-na-rinok-nerukhomosti-v-ukrayini-prognoz-na-2025-rik> (дата звернення: 19.10.2024).
8. Ринок нерухомості України: що буде з цінами, пропозицією і попитом. URL: <https://minfin.com.ua/ua/credits/articles/rynok-nedvizhimosti-ukrainy-chto-budet-s-cenami-predlozheniem-i-sprosom/>. (дата звернення: 01.11.2024).
9. Новаковська І.О. Управління міським землекористуванням: монографія. — К.: Аграр. наука, 2016. — 304 с.

10. Гербич Л.А. Методи переоцінювання заставного майна за іпотечними кредитами банку //Галицький економічний вісник. – 2014. - № 2 (45). – С. 93 – 101.
11. Максишко Н. К., Шаповалова В. О. Нечітка модель ідентифікації фаз на ринку нерухомості. Нейро-нечіткі технології моделювання в економіці. 2014. № 3. С. 94–119.
12. Григорків В. С., Ярошенко О. І., Філіпчук Н. В. Нейронні мережі та їх використання для прогнозування тенденцій ринку нерухомості // Науковий вісник НЛТУ України. 2012. – Вип. 22.5. – С. 324-330.
13. Калиніченко Ю.В., Добровольська В.В., Абрамчк А.Р. Моделювання тенденцій ринку нерухомості. / Ю.В. Калиніченко, В.В. Добровольська, А.Р. Абрамчк // Економічний аналіз: зб. наук. праць Тернопільського національного економічного університету, 2016. Том 23. № 1. – С. 28-36.
14. Machine Learning, ML. URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>. (date of access: 02.11.2024).
15. Хлевна Ю.Л., Бура Ю.С. Інформаційне забезпечення прогнозування цін на нерухомість методами машинного навчання. / Ю.Л. Хлевна, Ю.С. Бура // Sciences of Europe, 2021. - № 71. – Р. 54-62.
16. Predicting House Prices with Support Vector Regression in Python. URL: <https://soumenatta.medium.com/predicting-house-prices-with-support-vector-regression-in-python-46ce87686eb7>. (date of access: 02.11.2024).
17. Wan, H. Chinese Housing Prices Prediction Using Autoregressive Integrated Moving Average Model. HSET 2024. URL: [https://www.researchgate.net/publication/381543750\\_Chinese\\_Housing\\_Prices\\_Prediction\\_using\\_Autoregressive\\_Integrated\\_Moving\\_Average\\_Model](https://www.researchgate.net/publication/381543750_Chinese_Housing_Prices_Prediction_using_Autoregressive_Integrated_Moving_Average_Model). (date of access: 04.11.2024).
18. Manasa, J., Gupta, R., & Narahari, N. S. (2020). Machine learning based predicting house prices using regression techniques. DOI:[10.1109/ICIMIA48430.2020.9074952](https://doi.org/10.1109/ICIMIA48430.2020.9074952)

19. Воронін В. О., Мамчин М. М., Лянце Е. В. Прогнозне моделювання тенденцій розвитку ринку нерухомості // Вісник Національного університету "Львівська політехніка". – 2012. – № 735 : Логістика. – С. 38-46.
20. VanderPlas, J. Python Data Science Handbook: Essential Tools for Working with Data. — O'Reilly Media, 2017. — 546 с.
21. What is Ridge Regression. URL: <https://www.ibm.com/topics/ridge-regression>. (date of access: 06.11.2024).
22. Jeremia, N. E. ; Nurrohmah, S. ; Fithriani, I. / Robust Ridge regression to solve a multicollinearity and outlier. In: Journal of Physics: Conference Series. 2020 ; Vol. 1442, No. 1.
23. XGBoost Mathematics Explained URL: <https://towardsdatascience.com/xgboost-mathematics-explained-58262530904a>. (date of access: 07.11.2024)
24. Мірошніченко І.В., Крашеніннікова О.В. Прогнозування ціни на нерухомість з використанням алгоритмів машинного навчання. Ефективна економіка. 2022. № 1. — URL: <http://www.economy.nayka.com.ua/?op=1&z=9890> (дата звернення: 05.11.2024). DOI: 10.32702/2307-2105-2022.1.81
25. Nadeem. An introduction to decision tree. URL: <https://medium.com/analytics-vidhya/an-introduction-to-decision-tree-907ee016dce2>. (date of access: 08.11.2024)
26. Random forest. Вікіпедія: Вільна Енциклопедія. URL: [https://uk.wikipedia.org/wiki/Random\\_forest](https://uk.wikipedia.org/wiki/Random_forest). (дата звернення: 09.11.2024)
27. Hey, T., Tansley, S., Tolle, K. M., et al. The Fourth Paradigm: Data-Intensive Scientific Discovery. — Microsoft research Redmond, WA, 2009. — volume 1.
28. Donoho, D. 50 years of data science // Based on a Presentation at the Tukey Centennial Workshop. NJ Princeton, 2015.
29. Прийняття рішень на основі даних: як це може змінити ваші криві зростання. URL: <https://www.ranktracker.com/uk/blog/data-driven-decision-making-how-it-can-shift-your-growth-curves>. (дата звернення: 12.11.2024).

30. Data-Driven Decision Making: A Primer for Beginners. URL: <http://surl.li/hegclt>. (date of access: 11.11.2024).
31. The Evolution of the Data-Driven Company. URL: <https://www.christopherspenn.com/2019/08/the-evolution-of-the-data-driven-company/>. (date of access: 13.11.2024).
32. Чим відрізняються кількісні та якісні дані в маркетингу: на що спиратися при створенні дослідження? URL: <https://web-promo.ua/ua/blog/chim-vidriznyayutsya-kilkisni-ta-yakisni-dani-v-marketingu-na-sho-spiratisya-pri-stvorenni-doslidzhennya/>. (дата звернення: 15.11.2024).
33. Гавриленко С. Ю. Дослідження методів виявлення аномалій на етапі попередньої обробки даних / С. Ю. Гавриленко, В. Д. Зозуля // Системи управління, навігації та зв'язку = Control, navigation and communication systems : зб. наук. пр. / ред. В. В. Косенко ; Нац. ун-т "Полтав. політехніка ім. Юрія Кондратюка". – Полтава : Петров В. В., 2022. – Вип. 1 (67). – С. 52-56.
34. Як і для чого використовувати візуалізацію даних? URL: <http://eidos.org.ua/novyny/yak-i-dlya-choho-vykorystovuvaty-vizualizatsiyu-danyh/>. (дата звернення: 15.11.2024)
35. Data-driven decisions: a practical toolkit for library and information professionals. London: Facet Publishing, 2022. 180 с. ISBN 978-1-78330-478-3.
36. О.І. Кушлик-Дивульська, Б.Р. Кушлик. Основи теорії прийняття рішень. – К., 2014. – 94с.
37. Братушка С. М., Новак С. М., Хайлук С. О. Системи підтримки прийняття рішень : навчальний посібник для самостійного вивчення дисципліни. — Суми: ДВНЗ “УАБС НБУ”, 2010. — 265 с.
38. Захарчук В.І., Захарчук О.В. Методи системного аналізу об'єктів автомобільного транспорту: навчальний посібник для студентів та аспірантів вищих навчальних закладів. — Луцьк: РВВ Луцького НТУ, 2023. — 133 с.
39. Горда О. В. Теорія прийняття рішень: конспект лекцій. — Київ: КНУБА, 2023. — 120 с.

40. Як data-driven підхід допомагає ритейлерам масштабувати бізнес?  
URL: <https://datawiz.io/uk/blog/data-driven-approach-in-retailers>. (дата звернення: 15.11.2024)
41. Machine Learning and Data Science / ed. by P. Agrawal et al. Wiley, 2022.  
URL: <https://doi.org/10.1002/9781119776499> (date of access: 15.11.2024).
42. Types of Machine Learning - TechGuruSpeaks. *TechGuruSpeaks*.  
URL: <https://www.techguruspeaks.com/types-of-machine-learning/> (date of access: 15.11.2024).
43. Vasques X. Machine Learning Theory and Applications: Hands-On Use Cases with Python on Classical and Quantum Machines. Wiley & Sons, Limited, John, 2024.
44. Humantsattabhayya. All about Supervised Learning. *Medium*.  
URL: <https://medium.com/@humantsattabhayya/all-about-supervised-learning-3ef9cd5efe06> (date of access: 16.11.2024).
45. Grigorev A. Machine Learning Bookcamp. Manning Publications Co. LLC, 2022.
46. Pajankar A., Joshi A. Hands-On Machine Learning with Python: Implement Neural Network Solutions with Scikit-Learn and Pytorch. Apress L. P., 2022.
47. Contributors to Wikimedia projects. Neural network (machine learning) - Wikipedia. *Wikipedia, the free encyclopedia*.  
URL: [https://en.wikipedia.org/wiki/Neural\\_network\\_\(machine\\_learning\)](https://en.wikipedia.org/wiki/Neural_network_(machine_learning)) (date of access: 16.11.2024).
48. Що таке нейронні мережі і як вони працюють - Root-Nation.com. *Root-Nation.com*. URL: <https://root-nation.com/ua/articles-ua/tech-ua/ua-what-is-neural-network/> (дата звернення: 16.11.2024).
49. History | Твоя бібліотека. *History | Твоя електронна бібліотека*.  
URL: [https://uahistory.co/pidruchniki/zadorozhnij-biology-deep-level-8-class-2021/39.php#google\\_vignette](https://uahistory.co/pidruchniki/zadorozhnij-biology-deep-level-8-class-2021/39.php#google_vignette) (дата звернення: 16.11.2024).

50. Miller L. Python for Data Analysis: A Step by Step Guide to Build Intelligent System Machine Learning, Scikit-Learn, Keras and Tensorflow. Independently Published, 2020.

51. Ansari S. Building Computer Vision Applications Using Artificial Neural Networks. Berkeley, CA : Apress, 2023. URL: <https://doi.org/10.1007/978-1-4842-9866-4> (date of access: 16.11.2024).

52. Contributors to Wikimedia projects. Feedforward neural network - Wikipedia. *Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Feedforward\\_neural\\_network](https://en.wikipedia.org/wiki/Feedforward_neural_network) (date of access: 16.11.2024).

53. Принципи побудови нейронних мереж – IT Master - електроніка та програмування. *Головна – IT Master - електроніка та програмування*. URL: <https://itmaster.biz.ua/programming/vision/neural-networks-principles.html> (дата звернення: 16.11.2024).

54. What Is Gradient Descent? | Built In. *Built In*. URL: <https://builtin.com/data-science/gradient-descent> (date of access: 17.11.2024).

55. Навчання нейромережі класифікації зображень – IT Master - електроніка та програмування. *Головна – IT Master - електроніка та програмування*. URL: <https://itmaster.biz.ua/programming/vision/ne-study-classification.html> (дата звернення: 17.11.2024).

56. ML | Stochastic Gradient Descent (SGD) - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/> (date of access: 17.11.2024).

57. Soni B. Understanding the AdaGrad Optimization Algorithm: An Adaptive Learning Rate Approach. *Medium*. URL: [https://medium.com/@brijesh\\_soni/understanding-the-adagrad-optimization-algorithm-an-adaptive-learning-rate-approach-9dfaae2077bb](https://medium.com/@brijesh_soni/understanding-the-adagrad-optimization-algorithm-an-adaptive-learning-rate-approach-9dfaae2077bb) (date of access: 17.11.2024).

58. Metalearning / P. Brazdil et al. Cham: Springer International Publishing, 2022. URL: <https://doi.org/10.1007/978-3-030-67024-5> (date of access: 02.12.2024).

59. Hutter F. Automated Machine Learning: Methods, Systems, Challenges. Cham : Springer Nature, 2019. 219 p.

60. Sobrecueva L. Automated Machine Learning with AutoKeras: Deep learning made accessible for everyone with just few lines of coding. Packt Publishing, 2021. 194 p.

61. Label Encoding in Python - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/> (date of access: 18.11.2024).

62. One Hot Encoding Explained | Built In. *Built In*. URL: <https://builtin.com/articles/one-hot-encoding> (date of access: 18.11.2024).

63. Natural Language Toolkit – Вікіпедія. *Вікіпедія*. URL: [https://uk.wikipedia.org/wiki/Natural\\_Language\\_Toolkit](https://uk.wikipedia.org/wiki/Natural_Language_Toolkit) (дата звернення: 18.11.2024).

64. Klein E., Bird S., Loper E. Natural Language Processing with Python. O'Reilly Media, Incorporated, 2017.

65. AlShammari A. F. Implementation of Keyword Extraction using Term Frequency-Inverse Document Frequency (TF-IDF) in Python. *International Journal of Computer Applications*. 2023. Vol. 185, no. 35. P. 9–14. URL: <https://doi.org/10.5120/ijca2023923137> (date of access: 19.11.2024).

66. TfidfVectorizer. *scikit-learn*. URL: [https://scikit-learn.org/1.5/modules/generated/sklearn.feature\\_extraction.text.TfidfVectorizer.html](https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html) (date of access: 20.11.2024).

67. Daly Q. Step-by-Step NMF Example in Python. *Medium*. URL: <https://medium.com/@quindaly/step-by-step-nmf-example-in-python-9974e38dc9f9> (date of access: 20.11.2024).

68. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction – umap 0.5 documentation. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction – umap 0.5 documentation*. URL: <https://umap-learn.readthedocs.io/en/latest/> (date of access: 20.11.2024).

69. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd Edition Publisher: O'Reilly Media, 2017. 547 p.
70. Pandas - Python Data Analysis Library. *pandas - Python Data Analysis Library*. URL: <https://pandas.pydata.org/> (date of access: 21.11.2024).
71. MinMaxScaler. *scikit-learn*. URL: <https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.MinMaxScaler.html> (date of access: 21.11.2024).
72. What is Ridge Regression? -GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/what-is-ridge-regression/> (date of access: 21.11.2024).
73. Scikit-learn. From Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/Scikit-learn>. (Accessed: 17.03.2023).
74. Peter Bruce, Andrew Bruce, Peter Gedeck. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python, O'Reilly Media, 2020. 368 p.
75. Mean absolute error (MAE). Вікіпедія: Вільна Енциклопедія. URL: [https://computersciencewiki.org/index.php/Mean\\_absolute\\_error\\_\(MAE\)](https://computersciencewiki.org/index.php/Mean_absolute_error_(MAE)). (Accessed: 22.11.2024).
76. Jim Frost. Mean squared error (MSE). URL: <https://statisticsbyjim.com/regression/mean-squared-error-mse>. (Accessed: 21.11.2024).
77. Keras documentation. *Keras: Deep Learning for humans*. URL: <https://keras.io/api/layers/> (date of access: 23.11.2024).
78. Matplotlib 3.7.1 documentation. URL: <https://matplotlib.org/stable/index>. (date of access: 23.11.2024).
79. Keras documentation: KerasTuner. *Keras: Deep Learning for humans*. URL: [https://keras.io/keras\\_tuner/](https://keras.io/keras_tuner/) (date of access: 23.11.2024).
80. AutoKeras. *AutoKeras*. URL: <https://autokeras.com/> (date of access: 23.11.2024)

## ДОДАТКИ

```
!pip install autokeras
!pip install translate
!pip install pymorphy2
import warnings
import numpy as np
import pandas as pd
import gdown
import autokeras as ak
import tensorflow as tf
import keras
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import string
import requests
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras import utils
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from IPython.display import clear_output
from ipywidgets import interact, widgets
from keras.models import Sequential
from keras.optimizers import SGD
from keras.optimizers import RMSprop
from keras.optimizers import Adagrad
from keras.optimizers import Adam
from keras.layers import Dense
```

```

from keras.activations import leaky_relu
from keras.layers import Dense, LeakyReLU
from sklearn.feature_extraction.text import TfidfVectorizer
from pymorphy2 import MorphAnalyzer
from sklearn.preprocessing import normalize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from translate import Translator
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
nltk.download('wordnet')
clear_output()

# Завантаження файлу с даними
file_path = flats.xlsx'
df = pd.read_excel(file_path, sheet_name='Sheet1')
df = df.loc[~df.duplicated()]

# Знаходження середньої ціни по районах
average_price_by_settlement = df.groupby('settlement')['price'].mean().sort_values()

# Заміна значень району, на відповідне число, в залежності від середньої ціни по району
settlement_mapping = {settlement: idx for idx, settlement in
enumerate(average_price_by_settlement.index, 1)}
df['settlement'] = df['settlement'].map(settlement_mapping)

# Заміна категоріального признаку (тип санвузла) в числовий
bathroom_mode = df["bathroom"].mode()
df["bathroom"] = df["bathroom"].replace(np.nan, bathroom_mode[0])

bathroom_mapping = {
    'Суміжний': 2,

```

```

    'Роздільний': 3,
    '2 і більше': 4,
    'Санвузол відсутній': 1
}

```

```

df['bathroom'] = df['bathroom'].replace(bathroom_mapping)
# Заміна категоріального признаку (наявність меблювання) в числовий
furnishing_mode = df["furnishing"].mode()
df["furnishing"] = df["furnishing"].replace(np.nan, furnishing_mode[0])

```

```

furnishing_mapping = {
    'Ні': 0,
    'Так': 1
}

```

```

df['furnishing'] = df['furnishing'].replace(furnishing_mapping)
# Форматування стовпця площа квартири
df['area'] = df['area'].str.replace(' м²', '').astype(float)
# Форматування стовпця площа кухні квартири
df['kitchen_area'] = df['kitchen_area'].str.replace(' м²', '').astype(float)
# Заміна категоріального признаку (кількість кімнат) в числовий
number_of_rooms_mode = df["number_of_rooms"].mode()
df["number_of_rooms"] = df["number_of_rooms"].replace(np.nan,
number_of_rooms_mode[0])

```

```

rooms_mapping = {
    '1 кімната': 1,
    '2 кімнати': 2,
    '3 кімнати': 3,
    '4 кімнати': 4,

```

```

    '5+ кімнат': 5
}

df['number_of_rooms'] = df['number_of_rooms'].replace(rooms_mapping)
# Заміна категоріального признаку (тип квартири) в числовий
class_of_housing_mode = df["class_of_housing"].mode()
df["class_of_housing"] = df["class_of_housing"].replace(np.nan,
class_of_housing_mode[0])

class_of_housing_mapping = {
    'Еліт': 4,
    'Бізнес': 3,
    'Комфорт': 2,
    'Економ': 1,
}

df['class_of_housing'] = df['class_of_housing'].replace(class_of_housing_mapping)
# Заміна категоріального признаку (наявність ремонту) в числовий
repair_mode = df["repair"].mode()
df["repair"] = df["repair"].replace(np.nan, repair_mode[0])

repair_mapping = {
    'Євроремонт': 5,
    'Авторський проект': 6,
    'Після будівельників': 1,
    'Під чистову обробку': 2,
    'Косметичний ремонт': 4,
    'Житловий стан': 3
}

```

```

df['repair'] = df['repair'].replace(repair_mapping)
# Заміна категоріального признаку (тип будинку) в числовий
house_type_mode = df["house_type"].mode()
df["house_type"] = df["house_type"].replace(np.nan, house_type_mode[0])

house_type_mapping = {
    'Житловий фонд 80-90-і': 7,
    'Житловий фонд 91-2000-і': 8,
    'Житловий фонд від 2021 р.': 11,
    'Совмін': 1,
    'Сталінка': 5,
    'Хрущовка': 4,
    'Царський будинок': 6,
    'Чешка': 3,
    'Гостинка': 2,
    'Житловий фонд 2001-2010-і': 9,
    'Житловий фонд 2011-2020-і': 10,
}

df['house_type'] = df['house_type'].replace(house_type_mapping)
# Заміна пропусків на моду в стовпці тип стін
walls_mode = df["type_of_walls"].mode()
df["type_of_walls"] = df["type_of_walls"].replace(np.nan, walls_mode[0])
# Заміна пропусків на моду в стовпці тип опалення
heating_mode = df["heating"].mode()
df["heating"] = df["heating"].replace(np.nan, heating_mode[0])
# Заміна категоріального признаку (тип стін) в числовий
type_of_walls = df.type_of_walls.unique()

```

```

numbers_type_of_walls = [i for i in range(1,8)]

type_of_walls = dict(zip(type_of_walls, numbers_type_of_walls))
df.type_of_walls.replace(to_replace=type_of_walls, inplace=True)
# Заміна категоріального признаку (тип опалення) в числовий
heating = df.heating.unique()
numbers_heating = [i for i in range(1,7)]

heating = dict(zip(heating, numbers_heating))
df.heating.replace(to_replace=heating, inplace=True)
# Побудова кореляційної матриці між факторами
def plot_correlation_heatmap(df, target_column):
    numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
    correlation = df[numeric_columns].corr().sort_values(by=target_column,
ascending=False)

    plt.figure(figsize=(10, 8))
    sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title(f'Кореляція {target_column}')
    plt.show()
plot_correlation_heatmap(df, target_column='price')
# Перевірка статистики по цільовій змінній ціни
pd.set_option('display.float_format', lambda x: '%.2f % x')
df['price'].describe()

# Ціна - площа розкид даних графік
plt.figure(figsize=(7,8))
plt.scatter(df['area'], df['price'],color='blue')
plt.ticklabel_format(style='plain', axis='y')
plt.xlabel('area', fontsize=18)
plt.ylabel('price', fontsize=18)

```

```

plt.grid(color='black')
# Видалення викидів
df.drop(df[df['area'] > 100].index, inplace=True)
df.drop(df[df['price'] > 9000000].index, inplace=True)
df.drop(df[df['price'] < 3000000].index, inplace=True)

# Функція нормалізації даних
def normalize_dataframe(df):
    df_normalized = df.copy()

    numeric_cols = df_normalized.select_dtypes(include=['float64',
'int64']).columns.drop("price")
    scaler = MinMaxScaler(feature_range=(0, 1))
    df_normalized[numeric_cols] = scaler.fit_transform(df_normalized[numeric_cols])

    scaler_price = MinMaxScaler(feature_range=(0, 1))
    df_normalized['price'] = scaler_price.fit_transform(df_normalized[['price']])
    normalized = True

    return df_normalized, scaler_price
# @title
def split_data(df):

    X_train, X_test, y_train, y_test = train_test_split(df.drop(columns=['price']),
df['price'], test_size=0.1, random_state=92)

    X_train_np = X_train.to_numpy()
    y_train_np = y_train.to_numpy()

    X_test_np = X_test.to_numpy()

```

```

y_test_np = y_test.to_numpy()

return X_train_np, X_test_np, y_train_np, y_test_np

df_categories_numbers_columns = df.drop(columns=['link', 'plan', 'multimedia',
'utilities', 'type_of_object', 'full_description', 'comfort', 'household_appliances'])

df_categories_numbers_columns_normalized, scaler_price =
normalize_dataframe(df_categories_numbers_columns)
# Розділення на тестову та тренувальну вибірки
X_train, X_test, y_train, y_test =
split_data(df_categories_numbers_columns_normalized)
def evaluate_model(auto_model, X_test, y_test, scaler_price, normalized):
    with warnings.catch_warnings():
        warnings.simplefilter("ignore")
        y_pred = auto_model.predict(X_test)

    if normalized:
        y_pred_original = scaler_price.inverse_transform(y_pred).flatten()
        y_test_original = scaler_price.inverse_transform(y_test.reshape(-1, 1)).flatten()
    else:
        y_pred_original = y_pred.flatten()
        y_test_original = y_test.flatten()

    rmse = round(((sum((y_test_original - y_pred_original)**2)/len(y_test))**(0.5), 10)
    mae = round(np.mean(np.abs(y_test_original - y_pred_original)), 10)
    mape = round(np.mean(np.abs((y_test_original+1 - y_pred_original) /
(y_test_original+1)))) * 100, 10)

```

```

print('RMSE', rmse)
print('MAE', mae)
print('MAPE', mape)

# linear активація, 1 шар, 100 епох, оптимізація стохастичний градієнтний спуск
model = Sequential([
    Dense(13, input_shape=(13,), activation='linear')
    , Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=100)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)

# relu активація, 1 шар, 100 епох, оптимізація стохастичний градієнтний спуск
model = Sequential([
    Dense(13, input_shape=(13,), activation='relu')
    , Dense(1, activation='relu')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=100)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)

# relu активація в схованих шарах, 5 шарів, 100 епох, оптимізація стохастичний
градієнтний спуск
model = Sequential([
    Dense(128, input_shape=(13,), activation='linear')
    , Dense(64, activation='relu')
    , Dense(32, activation='relu')
    , Dense(16, activation='relu')
    , Dense(8, activation='relu')
    , Dense(4, activation='relu')
    , Dense(1, activation='linear')
])

```

```

])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=100)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
# relu активація в схованих шарах, 5 шарів, 200 епох, оптимізація стохастичний
градієнтний спуск
model = Sequential([
    Dense(128, input_shape=(13,), activation='linear')
    , Dense(64, activation='relu')
    , Dense(32, activation='relu')
    , Dense(16, activation='relu')
    , Dense(8, activation='relu')
    , Dense(4, activation='relu')
    , Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
model.fit(X_train, y_train, epochs=200)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
# leaky relu активація в схованих шарах, 5 шарів, 200 епох, оптимізація
стохастичний градієнтний спуск
model = Sequential([
    Dense(128, input_shape=(13,), activation='linear')
    , Dense(64)
    , LeakyReLU(alpha=0.01)
    , Dense(32)
    , LeakyReLU(alpha=0.01)
    , Dense(16)
    , LeakyReLU(alpha=0.01)
    , Dense(8)
    , LeakyReLU(alpha=0.01)

```

```

, Dense(4)
, LeakyReLU(alpha=0.01)
, Dense(1, activation='linear')
])
model.compile(optimizer='sgd', loss='mse')
history = model.fit(X_train, y_train, epochs=200)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
for i, i_optim in enumerate([SGD()]):

    # Фіксовані ваги
    keras.utils.set_random_seed(1)

    model = Sequential([
        Dense(128, input_shape=(13,), activation='linear')
        , Dense(64)
        , LeakyReLU(alpha=0.01)
        , Dense(32)
        , LeakyReLU(alpha=0.01)
        , Dense(16)
        , LeakyReLU(alpha=0.01)
        , Dense(8)
        , LeakyReLU(alpha=0.01)
        , Dense(4)
        , LeakyReLU(alpha=0.01)
        , Dense(1, activation='linear')
    ])

    model.compile(optimizer=i_optim, loss='mse', metrics=['mae'])
    history = model.fit(X_train, y_train, epochs=500)
    evaluate_model(model, X_test, y_test, scaler_price, normalized= True)

```

```

mae_values = history.history['mae'][100:]
print(mae_values)
epochs = range(100, 350)
print(epochs)
models[i_optim.get_config()['name']] = model
name = model.optimizer.get_config()['name']

plt.plot(epochs, mae_values, label=f'MAE {name}')
plt.xlabel('MAE')
plt.ylabel('Epoch')
plt.legend()
# Оптимізатори
for i, i_optim in enumerate([Adagrad(learning_rate=0.01)]):

    # Фіксовані ваги
    keras.utils.set_random_seed(1)

    model = Sequential([
        Dense(128, input_shape=(13,), activation='linear')
        , Dense(64)
        , LeakyReLU(alpha=0.01)
        , Dense(32)
        , LeakyReLU(alpha=0.01)
        , Dense(16)
        , LeakyReLU(alpha=0.01)
        , Dense(8)
        , LeakyReLU(alpha=0.01)
        , Dense(4)
        , LeakyReLU(alpha=0.01)
        , Dense(1, activation='linear')
    ])

```

```

])

# Компіляція моделі з одним з оптимізаторів
model.compile(optimizer=i_optim, loss='mse', metrics=['mae'])
history = model.fit(X_train, y_train, epochs=350)
evaluate_model(model, X_test, y_test, scaler_price, normalized= True)
mae_values = history.history['mae'][100:]
print(mae_values)
epochs = range(100, 350)
print(epochs)

# записуємо обчислену модель в словарь
models[i_optim.get_config()['name']] = model
name = model.optimizer.get_config()['name']

plt.plot(epochs, mae_values, label=f'MAE {name}')
plt.xlabel('MAE')
plt.ylabel('Epoch')
plt.legend()

# Вибір тільки найбільш корельованих змінних
df_significant_params_normalized =
df_categories_numbers_columns_normalized.drop(
    columns=['furnishing', 'heating', 'repair', 'number of storeys', 'type_of_walls', 'floor']
)

X_train, X_test, y_train, y_test = split_data(df_significant_params_normalized)
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from kerastuner.tuners import RandomSearch
from sklearn.model_selection import KFold

```

```

# Функція для створення моделі
def build_model(hp):
    model = keras.Sequential()

    # Визначення гіперпараметрів для тюнінга
    hp_units_1 = hp.Int('units_1', min_value=32, max_value=256, step=32)
    hp_units_2 = hp.Int('units_2', min_value=16, max_value=128, step=16)
    hp_units_3 = hp.Int('units_3', min_value=8, max_value=64, step=8)
    hp_units_4 = hp.Int('units_4', min_value=4, max_value=32, step=4)
    hp_units_5 = hp.Int('units_5', min_value=2, max_value=16, step=2)

    # Додавання слоїв в модель
    model.add(layers.Dense(units=hp_units_1, activation='linear', input_shape=(7,)))
    model.add(layers.Dense(units=hp_units_2))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(units=hp_units_3))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(units=hp_units_4))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(units=hp_units_5))
    model.add(layers.LeakyReLU(alpha=0.01))
    model.add(layers.Dense(1, activation='linear'))
    model.compile(optimizer='sgd', loss='mse')
    return model

tuner = RandomSearch(
    build_model,
    objective='val_loss',
    max_trials=8,

```

```

    executions_per_trial=2,
    directory='project_yaroslav',
    project_name='diplom'
)
for train_index, val_index in kfold.split(X_train):
    tuner.search(X_train[train_index], y_train[train_index],
    epochs=200,
    validation_data=(X_train[val_index], y_train[val_index]))

# Отримання найкращої моделі
best_model = tuner.get_best_models(num_models=1)[0]

# Вивід інформації про архітектуру мережі
best_model.summary()
evaluate_model(best_model, X_test, y_test, scaler_price, normalized= True)
# Додавання до моделі текстових стовпців котрі містять інформацію про
наповнення об'єкта нерухомості електро приладам та іншим обладнанням

df[['comfort', 'multimedia', 'utilities', 'household_appliances']] = df[['comfort',
'multimedia', 'utilities', 'household_appliances']].fillna('unknown')

df_text_feature = df['link']

# One hot encoding можливих значень цих стовпців
for col in ['comfort', 'multimedia', 'utilities', 'household_appliances']:
    cv = CountVectorizer(tokenizer=lambda x: x.split(' '))
    new_features = pd.DataFrame(cv.fit_transform(df[col]).toarray(),
columns=cv.get_feature_names_out(), index=df.index)
    df_text_feature = pd.concat([df_text_feature, new_features], axis=1)
df_categories_numbers_text_columns = pd.merge(df, df_text_feature, on='link')

```

```

df_categorical_numbers_text_columns =
df_categories_numbers_text_columns.drop(columns=['link', 'plan', 'multimedia',
'utilities', 'type_of_object', 'full_description', 'comfort', 'household_appliances',
'unknown'])
X_train, X_test, y_train, y_test = split_data(df_categorical_numbers_text_columns)

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from kerastuner.tuners import RandomSearch
from sklearn.model_selection import KFold
from kerastuner.tuners import Hyperband

# Функція для створення моделі
def build_model(hp):
    model = keras.Sequential()

    # Визначення гіперпараметрів для тюнінга
    hp_units_1 = hp.Int('units_1', min_value=1500, max_value=3000, step=350)
    hp_units_2 = hp.Int('units_2', min_value=1000, max_value=2000, step=250)
    hp_units_3 = hp.Int('units_3', min_value=500, max_value=1000, step=150)
    hp_units_4 = hp.Int('units_4', min_value=250, max_value=500, step=50)
    hp_units_5 = hp.Int('units_5', min_value=50, max_value=100, step=20)

    # Додавання слоїв в модель
    model.add(layers.Dense(units=hp_units_1, activation='linear', input_shape=(62,)))
    model.add(layers.Dense(units=hp_units_2, activation='linear'))
    model.add(layers.Dense(units=hp_units_3, activation='linear'))
    model.add(layers.Dense(units=hp_units_4, activation='linear'))
    model.add(layers.Dense(units=hp_units_5, activation='linear'))

```

```

model.add(layers.Dense(1, activation='linear'))
model.compile(optimizer='sgd', loss='mse')
return model

tuner = Hyperband(
    build_model,
    objective='val_loss',
    max_epochs=20,
    directory='project_yaroslav17',
    project_name='diplom17'
)
for train_index, val_index in kfold.split(X_train):
    tuner.search(X_train[train_index], y_train[train_index],
        validation_data=(X_train[val_index], y_train[val_index]))

# Отримання найкращої моделі
best_model = tuner.get_best_models(num_models=1)[0]

# Вивід інформації про архітектуру мережі
best_model.summary()
evaluate_model(best_model, X_test, y_test, scaler_price, normalized = True)
df['full_description'] = df['full_description'].str[4:]
df['full_description'] = df['full_description'].apply(lambda x: re.sub(r'^а-яА-ЯііїеЄëË\s|', '', x, flags=re.IGNORECASE))
df['full_description'] = df['full_description'].apply(lambda x: re.sub(r'\b\w{1,6}\b|', '', x, flags=re.IGNORECASE))
df['full_description'] = df['full_description'].apply(split_camel_case)
df['full_description'] = df['full_description'].apply(lambda x: re.sub(r'\b\w{1,6}\b|', '', x, flags=re.IGNORECASE))
df['full_description'] = df['full_description'].str.lower()

```

```

# TF IDF техніка важливості слів в стопці опису об'єкта нерухомості
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
texts_tfidf = tfidf_vectorizer.fit_transform(df['full_description'])
words = tfidf_vectorizer.get_feature_names_out()
bag_of_words = pd.DataFrame(texts_tfidf.todense(), columns=words)
from sklearn.decomposition import NMF
nnmf = NMF(n_components=10)
nnmf_result = nnmf.fit_transform(bag_of_words)
from umap import UMAP
umap = UMAP(n_components=2)
umap_result = umap.fit_transform(nnmf_result)
umap_df = pd.DataFrame(umap_result, columns=['dim_1', 'dim_2'])
df = df.reset_index(drop = True)
dataframe = pd.concat([df, umap_df], axis=1)
# One hot encoding можливих значень цих стовпців
for col in ['comfort', 'multimedia', 'utilities', 'household_appliances']:
    cv = CountVectorizer(tokenizer=lambda x: x.split(', '))
    new_features = pd.DataFrame(cv.fit_transform(dataframe[col]).toarray(),
columns=cv.get_feature_names_out(), index=dataframe.index)
    df_text_feature = pd.concat([dataframe, new_features], axis=1)
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from kerastuner.tuners import BayesianOptimization
from sklearn.model_selection import KFold

# Функція для створення моделі
def build_model(hp):
    model = keras.Sequential()

```

```

# Визначення гіперпараметрів для тюнінгу
hp_units_1 = hp.Int('units_1', min_value=32, max_value=1024, step=32)
hp_units_2 = hp.Int('units_2', min_value=16, max_value=512, step=16)
hp_units_3 = hp.Int('units_3', min_value=8, max_value=64, step=8)
hp_units_4 = hp.Int('units_4', min_value=4, max_value=32, step=4)
hp_units_5 = hp.Int('units_5', min_value=2, max_value=16, step=2)

# Додавання шарів у модель
model.add(layers.Dense(units=hp_units_1, activation='linear', input_shape=(24,)))
model.add(layers.Dense(units=hp_units_2))
model.add(layers.LeakyReLU(alpha=0.01))
model.add(layers.Dense(units=hp_units_3))
model.add(layers.LeakyReLU(alpha=0.01))
model.add(layers.Dense(units=hp_units_4))
model.add(layers.LeakyReLU(alpha=0.01))
model.add(layers.Dense(units=hp_units_5))
model.add(layers.LeakyReLU(alpha=0.01))
model.add(layers.Dense(1, activation='linear'))
model.compile(optimizer='sgd', loss='mse')
return model

tuner = BayesianOptimization(
    build_model,
    objective='val_loss',
    max_trials=1,
    executions_per_trial=1,
    directory='project_yaroslav23',
    project_name='diplom23'
)

```

```

# Створення об'єкту KFold
kfold = KFold(n_splits=2, shuffle=True, random_state=42)

for train_index, val_index in kfold.split(X_train):
    tuner.search(X_train[train_index], y_train[train_index],
                 epochs=150,
                 validation_data=(X_train[val_index], y_train[val_index]))

# Отримання найкращої моделі
best_model = tuner.get_best_models(num_models=1)[0]

# Вивід інформації про архітектуру мережі
best_model.summary()
evaluate_model(best_model, X_test, y_test, scaler_price, normalized= True)
import autokeras as ak
import tensorflow as tf
from sklearn.model_selection import KFold

input_node = ak.Input()
output_node = ak.DenseBlock()(input_node)
output_node = ak.RegressionHead()(output_node)

auto_model = ak.AutoModel(inputs = input_node, outputs = output_node,
                           overwrite=True,
                           max_trials=10, directory='project_yaroslav29', project_name='diplom29')

kfold = KFold(n_splits=3, shuffle=True, random_state=42)

for train_index, val_index in kfold.split(X_train):

```

```
auto_model.fit(X_train[train_index], y_train[train_index], epochs=10,  
               validation_data=(X_train[val_index], y_train[val_index]))  
  
best_model = auto_model.export_model()  
best_model.summary()  
  
evaluate_model(best_model, X_test, y_test, scaler_price, normalized= True)
```



## Звіт подібності

## метадані

Заголовок:

Овчинников\_Катуніна\_плагіат

Автор:

Науковий керівник / Експерт

Овчинников

Катуніна

Підрозділ:

кафедра математичного моделювання та статистики

## Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

|                        |  |    |
|------------------------|--|----|
| Заміна букв            |  | 3  |
| Інтервали              |  | 0  |
| Мікропробіли           |  | 14 |
| Білі знаки             |  | 3  |
| Парафрази (SmartMarks) |  | 50 |

## Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

Довжина фрази для коефіцієнта подібності 2



КП 2

18520

Кількість слів



КЦ

141921

Кількість символів

## Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

| 10 найдовших фраз   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                 | Копію тексту |  |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|--------------|--|
| ПОРЯДКОВИЙ<br>НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ<br>(ФРАГМЕНТІВ) |              |  |
| 1                   | <a href="http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9E%D0%97%42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92_%D0%9C_%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%84%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%86%D0%B2.pdf">http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9E%D0%97%42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92_%D0%9C_%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%84%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%86%D0%B2.pdf</a> | 114                                             | 0.62 %       |  |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |    |        |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|--------|
| 2  | <a href="http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9F%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf">http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9F%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf</a> | 59 | 0.32 % |
| 3  | <a href="https://pravobadunblog.files.wordpress.com/2016/02/d0b2-d180-d0bad183d187d0b5d180d0b5d0bdd0bad0be-d0bed186d196d0bdd0bad0b0-d0b1d196d0b7d0bd0b5d181d183-d182d0b0-d0bdd0b5d180d183d185.pdf">https://pravobadunblog.files.wordpress.com/2016/02/d0b2-d180-d0bad183d187d0b5d180d0b5d0bdd0bad0be-d0bed186d196d0bdd0bad0b0-d0b1d196d0b7d0bd0b5d181d183-d182d0b0-d0bdd0b5d180d183d185.pdf</a>                                                                                                                                                                                                                                                                                                                                                                             | 57 | 0.31 % |
| 4  | <a href="http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9F%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf">http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9F%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf</a> | 54 | 0.29 % |
| 5  | <a href="https://pravobadunblog.files.wordpress.com/2016/02/d0b2-d180-d0bad183d187d0b5d180d0b5d0bdd0bad0be-d0bed186d196d0bdd0bad0b0-d0b1d196d0b7d0bd0b5d181d183-d182d0b0-d0bdd0b5d180d183d185.pdf">https://pravobadunblog.files.wordpress.com/2016/02/d0b2-d180-d0bad183d187d0b5d180d0b5d0bdd0bad0be-d0bed186d196d0bdd0bad0b0-d0b1d196d0b7d0bd0b5d181d183-d182d0b0-d0bdd0b5d180d183d185.pdf</a>                                                                                                                                                                                                                                                                                                                                                                             | 52 | 0.28 % |
| 6  | <a href="http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9F%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf">http://dspace.wunu.edu.ua/bitstream/316497/34294/1/%D0%95%D0%9F%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf</a> | 44 | 0.24 % |
| 7  | <a href="https://ir.kneu.edu.ua/bitstreams/151df6cf-1099-4401-925c-5a0fe3423590/download">https://ir.kneu.edu.ua/bitstreams/151df6cf-1099-4401-925c-5a0fe3423590/download</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 35 | 0.19 % |
| 8  | <a href="https://ir.kneu.edu.ua/bitstreams/151df6cf-1099-4401-925c-5a0fe3423590/download">https://ir.kneu.edu.ua/bitstreams/151df6cf-1099-4401-925c-5a0fe3423590/download</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 32 | 0.17 % |
| 9  | <a href="https://openarchiv.nure.ua/server/api/core/bitstreams/b6649e08-a147-4ca6-849f-45cb518321a6/content">https://openarchiv.nure.ua/server/api/core/bitstreams/b6649e08-a147-4ca6-849f-45cb518321a6/content</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 25 | 0.13 % |
| 10 | <a href="https://docplayer.net/73642167-Informaciyno-analitichne-zabezpechennya-finansovekonomichnyi-diyalnosti-agrarnih-pidpriemstv-v-suchasnih-umovah.html">https://docplayer.net/73642167-Informaciyno-analitichne-zabezpechennya-finansovekonomichnyi-diyalnosti-agrarnih-pidpriemstv-v-suchasnih-umovah.html</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 21 | 0.11 % |

#### з бази даних RefBooks (0.06 %)

| ПОРЯДКОВИЙ<br>НОМЕР | ЗАГОЛОВОК                                                                                                                                                                                            | КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ<br>(ФРАГМЕНТІВ) |        |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|--------|
| джерело: Paperity   |                                                                                                                                                                                                      |                                                 |        |
| 1                   | Analysis of application a methods of machine learning based on artificial neural networks in the tasks of detecting cybersecurity threats<br>Herman Zastelo, Yevhen Shpachinskiy, Andrii Shevchenko; | 11 (1)                                          | 0.06 % |

#### з домашньої бази даних (0.15 %)

| ПОРЯДКОВИЙ<br>НОМЕР | ЗАГОЛОВОК                                                                                                                                                                                                              | КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ<br>(ФРАГМЕНТІВ) |        |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|--------|
| 1                   | <a href="#">Аналіз інвестиційного портфелю на основі прогнозування курсів акцій</a><br>12/16/2021<br>Kyiv National Economic University named after Vadym Hetman KNEU (кафедра математичного моделювання та статистики) | 16 (1)                                          | 0.09 % |
| 2                   | <a href="#">Прогнозування ціни на пальне в Україні</a><br>12/16/2022<br>Kyiv National Economic University named after Vadym Hetman KNEU (кафедра математичного моделювання та статистики)                              | 11 (1)                                          | 0.06 % |

## з програми обміну базами даних (0.41 %)

| ПОРЯДКОВИЙ<br>НОМЕР | ЗАГОЛОВОК                                                                                                                                                                                          | КІЛЬКІСТЬ (ІДЕНТИФІКАЦІЙНИХ СЛІВ<br>(ФРАГМЕНТІВ)) |        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|--------|
| 1                   | Дослідження методів прогнозування попиту на сільськогосподарську продукцію на основі аналізу наявних даних<br>12/11/2023<br>Dniprovsk State Technical University (Програмного забезпечення систем) | 16 (2)                                            | 0.09 % |
| 2                   | ФАЕТ_151_AKIK_2024_Петров_ДМ<br>7/10/2024<br>Ukrainian national aviation university (Ukrainian national aviation university)                                                                       | 16 (2)                                            | 0.09 % |
| 3                   | Дослідження моделі системи пошуку та підбору людей за визначеними критеріями<br>12/25/2023<br>State University of Telecommunications (NHIT)                                                        | 13 (1)                                            | 0.07 % |
| 4                   | Розроблення та машинне навчання нейронної моделі для аналізу медичних даних<br>10/28/2024<br>University of Customs and Finance course papers (University of Customs and Finance course papers)     | 13 (1)                                            | 0.07 % |
| 5                   | Розроблення інтерактивного застосунку оцінювання рівня енергетичної безпеки<br>5/29/2023<br>Kharkiv National University of Economics named after S. Kuznets (KNUE) (KNUE)                          | 12 (1)                                            | 0.06 % |
| 6                   | НЕЙРОМЕРЕЖЕВІ МОДЕЛІ ОБРОБКИ ТЕКСТУ<br>12/3/2023<br>V. N. Karazin Kharkiv National University (KGNU) (Факультет комп'ютерних наук - кафедра теоретичної та прикладної системотехніки)              | 6 (1)                                             | 0.03 % |

## з Інтернету (4.24 %)

| ПОРЯДКОВИЙ<br>НОМЕР | ДЖЕРЕЛО URL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | КІЛЬКІСТЬ (ІДЕНТИФІКАЦІЙНИХ СЛІВ<br>(ФРАГМЕНТІВ)) |        |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|--------|
| 1                   | <a href="http://dspace.wunu.edu.ua/jspui/bitstream/316497/34294/1/%D0%95%D0%9E%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf">http://dspace.wunu.edu.ua/jspui/bitstream/316497/34294/1/%D0%95%D0%9E%D0%97-42%20%D0%93%D0%B5%D0%B2%D0%B0%D0%BA%20%D0%92.%D0%9C.%D0%9D%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C%20%D1%8F%D0%BA%20%D0%BE%D0%B1%D1%94%D0%BA%D1%82%20%D0%B5%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D1%96%D1%87%D0%BD%D0%B8%D1%85%20%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%B5%D1%81%D1%96%D0%B2.pdf</a> | 282 (5)                                           | 1.52 % |
| 2                   | <a href="https://pravobadunblog.files.wordpress.com/2016/02/d0b2-d180-d0bad183d187d0b5d180d0b5d0bdd0bad0be-d0bed186d196d0bdd0bad0b0-d0b1d196d0b7d0bdd0b5d181d183-d182d0b0-d0bd0b5d180d183d185.pdf">https://pravobadunblog.files.wordpress.com/2016/02/d0b2-d180-d0bad183d187d0b5d180d0b5d0bdd0bad0be-d0bed186d196d0bdd0bad0b0-d0b1d196d0b7d0bdd0b5d181d183-d182d0b0-d0bd0b5d180d183d185.pdf</a>                                                                                                                                                                                                                                                                                                                                                                                         | 109 (2)                                           | 0.59 % |
| 3                   | <a href="https://openarchive.nure.ua/server/api/core/bitstreams/b6649e08-a147-4ca6-849f-45cb518321a6/content">https://openarchive.nure.ua/server/api/core/bitstreams/b6649e08-a147-4ca6-849f-45cb518321a6/content</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 86 (5)                                            | 0.46 % |
| 4                   | <a href="https://ir.kneu.edu.ua/bitstreams/151df6cf-1099-4401-925c-5a0fe3423590/download">https://ir.kneu.edu.ua/bitstreams/151df6cf-1099-4401-925c-5a0fe3423590/download</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 78 (3)                                            | 0.42 % |
| 5                   | <a href="https://termin.in.ua/nejromerezha/">https://termin.in.ua/nejromerezha/</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 63 (4)                                            | 0.34 % |
| 6                   | <a href="https://roslavslavmelnyk.com.ua/2024/11/14/vplyv-vijny-na-rynok-neruhomosti-v-ukrayini-prognoz-na-2025-rik/">https://roslavslavmelnyk.com.ua/2024/11/14/vplyv-vijny-na-rynok-neruhomosti-v-ukrayini-prognoz-na-2025-rik/</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 50 (4)                                            | 0.27 % |
| 7                   | <a href="https://ua-referat.com/uploaded/analitika-danih-v-procesi-oriynvatya-upravlinsekih-risheh/index1.html">https://ua-referat.com/uploaded/analitika-danih-v-procesi-oriynvatya-upravlinsekih-risheh/index1.html</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 22 (2)                                            | 0.12 % |
| 8                   | <a href="https://docplayer.net/73642167-Informacijno-analitichne-zabezpechennya-finansovoekonomichnoyi-diyalnosti-agrarnih-pidpriemstv-v-suchasnih-umovah.html">https://docplayer.net/73642167-Informacijno-analitichne-zabezpechennya-finansovoekonomichnoyi-diyalnosti-agrarnih-pidpriemstv-v-suchasnih-umovah.html</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 21 (1)                                            | 0.11 % |
| 9                   | <a href="http://www.library.univ.kiev.ua/ukr/host/viking/db/ftp/univ/zn_p_v_knu/zn_p_v_knu_2017_58.pdf">http://www.library.univ.kiev.ua/ukr/host/viking/db/ftp/univ/zn_p_v_knu/zn_p_v_knu_2017_58.pdf</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 18 (2)                                            | 0.10 % |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |        |        |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------|
| 10 | <a href="https://openarchive.nure.ua/server/api/core/bitstreams/ae44ba9f-ed80-459a-b634-62580dd308eb/content">https://openarchive.nure.ua/server/api/core/bitstreams/ae44ba9f-ed80-459a-b634-62580dd308eb/content</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 16 (3) | 0.09 % |
| 11 | <a href="https://phm.cuspu.edu.ua/ojs/index.php/SNYS/article/download/2037/pdf">https://phm.cuspu.edu.ua/ojs/index.php/SNYS/article/download/2037/pdf</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 15 (1) | 0.08 % |
| 12 | <a href="http://lib.ndu.edu.ua/dspace/bitstream/123456789/3382/1/%D0%A0%D0%BE%D0%B7%D0%B2%D0%B8%D1%82%D0%BE%D0%BA%20%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BF%D0%B3%D0%BE%20%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%82%D0%B0%20%D0%B9%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%B2%20%D1%80%D1%96%D0%B7%D0%BD%D0%B8%D1%85%20%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D1%85%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B0%D1%85%20%D1%96%20%D0%BD%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B8%D1%85%20%D0%B4%D0%BE%D1%81%D0%B8%D1%96%D0%B4%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F%D1%85_d.pdf">http://lib.ndu.edu.ua/dspace/bitstream/123456789/3382/1/%D0%A0%D0%BE%D0%B7%D0%B2%D0%B8%D1%82%D0%BE%D0%BA%20%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BF%D0%B3%D0%BE%20%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%82%D0%B0%20%D0%B9%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%B2%20%D1%80%D1%96%D0%B7%D0%BD%D0%B8%D1%85%20%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D1%85%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B0%D1%85%20%D1%96%20%D0%BD%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B8%D1%85%20%D0%B4%D0%BE%D1%81%D0%B8%D1%96%D0%B4%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F%D1%85_d.pdf</a> | 14 (1) | 0.08 % |
| 13 | <a href="http://bulletin-econom.univ.kiev.ua/wc-content/plugins/download-attachments/includes/download.php?id=6126">http://bulletin-econom.univ.kiev.ua/wc-content/plugins/download-attachments/includes/download.php?id=6126</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 7 (1)  | 0.04 % |
| 14 | <a href="https://www.ewdtest.com/asu/wc-content/uploads/2015/05/%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D1%8B-27-32-%D0%B8%D0%B7-ASU_166_2014.pdf">https://www.ewdtest.com/asu/wc-content/uploads/2015/05/%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D1%8B-27-32-%D0%B8%D0%B7-ASU_166_2014.pdf</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 5 (1)  | 0.03 % |

#### Список принятых фрагментів (немає принятих фрагментів)

|                  |       |                                       |
|------------------|-------|---------------------------------------|
| ПОРЯДКОВИЙ НОМЕР | ЗМІСТ | КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|-------|---------------------------------------|

**Овчинников Я.С.**

**Електронна пошта: yaroslavovchinnikov29@gmail.com**

**Науковий керівник : к.е.н, доцент Катуніна О.С.**

**Електронна пошта: katunina.olga@kneu.edu.ua**

*Київський національний економічний університет імені Вадима Гетьмана*

## **DATA DRIVEN ПІДХІД В ПРОГНОЗУВАННІ ЦІНИ НА ОБ'ЄКТИ НЕРУХОМОСТІ УКРАЇНИ**

**Ovchynnykov Y.S**

### **DATA DRIVEN APPROACH IN FORECASTING REAL ESTATE PRICES IN UKRAINE**

*У роботі досліджено методи прогнозування ціни на об'єкти житлової нерухомості в Україні з застосуванням дата-орієнтованого підходу. Для аналізу та передбачення ціни на ринку нерухомості побудовано низку моделей машинного навчання, зокрема, на базі нейромережових технологій.*

*The work examines the methods of forecasting prices for residential real estate in Ukraine using data-driven methods. To analyze and predict prices in the real estate market, a number of machine learning models have been built, in particular, based on neural network technologies.*

Прогнозування цін на ринку нерухомості України вимагає комплексного аналізу великих обсягів даних. Дата-орієнтований підхід дозволяє збирати та опрацьовувати інформацію про характеристики нерухомості, історичні ціни та на основі великих обсягів даних прогнозувати ціни на житлові об'єкти, відтворювати тенденції та закономірності формування відповідного ринку, а в подальшому - параметри його місткості. Для ідентифікації взаємозв'язків між великою кількістю предикторів, що охоплюють деталізований опис житлових пропозицій найбільш ефективним є модельний інструментарій машинного навчання. Data-Driven-технології на базі Data Science/Machine Learning/Deep Learning дозволяють комплексно проаналізувати систему попиту-підприємств факторів, у тому числі латентних, що містяться як у структурованих табличних, так і неструктурованих даних – текстах та зображеннях, що супроводжують деталізований опис ринкової пропозиції.

Для прогнозування ціни на об'єкти житлової нерухомості, використано набір даних що складається з 5000 тисяч актуальних оголошень про продаж об'єктів нерухомості, що були зібрані за період з 1 вересня по 1 листопада 2024 року, з сайту нерухомості olx.ua [1]. Набір даних для аналізу містить цільову змінну ціни об'єкта, та інформацію про характеристики об'єкта (всього - 15 характеристик).

В процесі проведеного дослідження застосовано модель лінійної регресії, що показала такі показники ефективності на тестовому наборі даних:

- RMSE = 965197 грн.
- MAPE = 15.8 %
- MAE = 772516 грн.

Метрика RMSE вказує на те, що в середньому передбачені значення відрізняються від фактичних експериментальних даних на 965197 гривень. З іншого боку, MAPE показує, що в середньому прогнозовані величини відхиляються від реальних даних на 15.8 %.

У процесі дослідження було застосовано техніку регуляризації, що є ефективним інструментом запобігання перенавчанню. Побудовано модель з використанням L2-регуляризації (Ridge-регресія) [3], одержано такі метрики ефективності:

- RMSE = 908643 грн.
- MAPE = 15 %
- MAE = 727181 грн.

Наступним етапом було нейромережеве моделювання для прогнозування ціни на об'єкти житлової нерухомості. На відміну від лінійних регресійних моделей, нейронні мережі здатні виявляти складні нелінійні залежності в даних завдяки своїй багатошаровій архітектурі та нелінійним функціям активації [4].

Нейронна мережа складається з 5 прихованих шарів та використовує функцію активації ReLU (Rectified Linear Unit). Для навчання моделі використано стохастичний градієнтний спуск – ітеративний алгоритм оптимізації, який мінімізує функцію втрат шляхом поступового оновлення ваг та зміщень у напрямку, протилежному градієнту функції втрат. Ці параметри отримані з застосуванням технології автоматизованого машинного навчання, котра дозволяє підібрати найкращі параметри, з урахуванням мінімізації помилки. За результатами моделювання, було отримано такі результати:

- RMSE = 845038.9 грн.
- MAPE = 13.95%
- MAE = 676478.3 грн.

Таким чином, у результаті дослідження з побудови та налаштування гіперпараметрів низки ML-моделей для прогнозування ціни нерухомості визначено, що найкращу якість прогнозу на тестовому наборі даних дають нейромережеві моделі, які за умови ретельного підбору архітектури та при збагаченні початкових даних додатковими ринковими характеристиками, зокрема, текстовими описами житла, дадуть можливість сформувати ефективну базу моделей для схожих застосувань у малому та середньому бізнесі.

#### *Література:*

1. Оголошення OLX.ua: сервіс оголошень України. URL: <https://www.olx.ua/d/uk/nedvizhimost/kvartiry/kyiv/>.
2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning with Applications in Python, 2023. 593p.
3. Andreas C. Müller, Sarah Guido. Introduction to Machine Learning with Python. O'Reilly Media, 2016. 366p.
4. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 801 p.

# СЕРТИФІКАТ

№ 11-2024/055

що засвідчує участь

## Ярослава Овчинникова

у XII ВСЕУКРАЇНСЬКІЙ НАУКОВО-ПРАКТИЧНІЙ КОНФЕРЕНЦІЇ  
ФОРУМІ МОЛОДИХ ЕКОНОМІСТІВ-КІБЕРНЕТИКІВ

«МОДЕЛЮВАННЯ ЕКОНОМІКИ: ПРОБЛЕМИ, ТЕНДЕНЦІЇ, ДОСВІД»

Професійна програма стажування  
12 годин (0,4 кредиту ECTS)



Декан економічного факультету  
Ростислав МИХАЙЛИШИН

22-23 листопада 2024 року  
м. Львів