
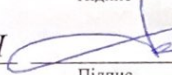



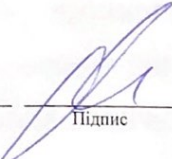
Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-19-2  А.О. Таланчук
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: к.т.н., доцент кафедри КН  О.А.Пасічник
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
Підпис Ініціали, прізвище

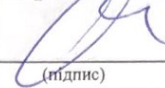
01 06 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор О.В. Бармак


« 6 » 03 2023 року


**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських»
2. Завдання видано студенту Таланчуку Андрію Олексійовичу
(прізвище, ім'я, по батькові)
3. Керівник роботи доцент кафедри КН Пасічник Олександр Анатолійович
(посада, прізвище, ім'я, по батькові)
4. Затверджено наказом університету від « 1 » 03 2023 р. № 5
5. Дата видачі завдання студенту: « 3 » 03 2023 р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані: Провести аналіз предметної області, аналіз сучасних способів передбачення ціни житла ріелторських системах; аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем; реалізувати інформаційну технологію способу передбачення ціни/собівартості житла засобами штучного інтелекту у розподілених ріелторських системах; провести експериментальне тестування інформаційної технології. Вихідними даними є інформація про оцінки складових собівартості житла з реальних систем предметної області.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	виконано
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	виконано
4	Робота над розділом 2 – Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах	березень 2023	виконано
5	Робота над розділом 3 – Програмна реалізація способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах	квітень 2023	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	виконано

конавець: студент 4 курсу, група КН-19-2  А.О. Таланчук
 Курс, група виконавця Підпис Ініціали, прізвище

івник: к.т.н., доцент кафедри КН  О.А. Пасічник
 Науковий ступінь, посада Підпис Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-2 Таланчук Андрій Олексійович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Пасічник Олександр Анатолійович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
65	47	0	30	2

Мета кваліфікаційної роботи полягає в створенні способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах. Для досягнення поставленої мети визначені наступні задачі дослідження: аналіз сучасних способів передбачення ціни житла ріелторських системах; аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем; реалізувати інформаційну технологію способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах; провести експериментальне тестування інформаційної технології.

Результатом виконання кваліфікаційної роботи бакалавра є розробка способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах та його реалізація як інформаційної системи.

Ключові слова: штучний інтелект, розподілені системи, передбачення ціни житла.

Виконавець: студент 4 курсу, група КН-19-2
Курс, група виконавця


Підпис

А.О. Таланчук
Ініціали, прізвище

Зміст

Перелік скорочень	4
Вступ.....	5
Розділ 1 Характеристика предметної області та постановка задачі	7
1.1 Аналіз предметної області	7
1.2 Аналіз сучасних способів передбачення ціни житла у ріелторських системах	9
1.3 Аналіз методів та засобів штучного інтелекту в задачах передбачення.....	10
1.4 Аналіз існуючих рішень для розподілених систем передбачення ціни житла в ріелторських установах	14
1.5 Мета, задачі та вимоги до реалізації способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах	16
Розділ 2 Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах	18
2.1 Загальна послідовність задачі передбачення	18
2.2 Загальна постановка задачі передбачення ціни житла.....	19
2.3 Загальна послідовність передбачення ціни житла.....	21
2.4 Особливості застосування засобів штучного інтелекту у розподілених ріелторських системах для передбачення ціни житла	24
2.5 Функціональна структура інформаційної системи.....	30
2.6 Проектування структури інформаційної системи	31
2.7 Висновки до розділу 2	34
Розділ 3 Програмна реалізація способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах	35
3.1 Структура модулів системи, їх взаємозв'язок	35

3.2 Особливості реалізації способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.....	37
3.3 Розробка програмних модулів	44
3.4 Валідація способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.....	57
3.5 Висновки до розділу 3	59
Висновки	61
Перелік посилань.....	63
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
PPC	Розподілена ріелторська система
ШІ	Штучний інтелект
API	Application Programming Interface
HDD	Hard Disk Drive
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Square Error
SSD	Solid State Drive

Вступ

Кваліфікаційна робота бакалавра присвячена розробці способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

Актуальність теми. Об'єктивний розвиток науки та технологій у поєднанні зі зростання можливостей технічних засобів визначили стрімкий розвиток методів та систем штучного інтелекту. Можливості методів штучного інтелекту, їх універсальність та ефективність обумовлюють їх широке впровадження у різноманітні сфери діяльності людини. Першочерговими є задачі пов'язані із прогнозування результатів діяльності, особливо коли мова йдеться про ресурсоємкі проекти із суттєвими різноманітними обмеженнями. Однією з найбільш складних та відповідальних напрямків діяльності, особливо після повномасштабного вторгнення в Україну, є побудова нового та відбудова зруйнованого житлового фонду. Величезні обсяги цієї задачі у поєднанні із об'єктивно обмеженими, фінансовими, товарними та часовими ресурсами обумовлюють необхідність ретельного планування цієї діяльності.

Мета і задачі роботи. Мета кваліфікаційної роботи полягає в створенні способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

Для досягнення поставленої мети визначені наступні задачі дослідження:

- аналіз сучасних способів передбачення ціни житла ріелторських системах;
- аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем;
- реалізувати інформаційну технологію способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах;
- провести експериментальне тестування інформаційної технології.

Об'єкт дослідження – процес побудови способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

Предмет дослідження – моделі, алгоритми та засоби для створення способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Предметною областю є передбачення засобами штучного інтелекту, а саме передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах. Для цього необхідно проаналізувати даний сегмент ринку нерухомості, а також поглянути на технології штучного інтелекту, які використовуються для передбачення цін.

Одним з основних аспектів, який потрібно вивчити, є те, як ріелторські системи працюють з цінами на нерухомість. Розглянемо, як зазвичай встановлюються ціни на житло. Зазвичай, ціни встановлюються на основі відомостей про ринок нерухомості, таких як тип нерухомості, її розташування, розмір, рік будівництва, стан будівлі та багато інших факторів. Також у встановленні цін на нерухомість важливу роль відіграють експерти, які оцінюють ціну на основі свого досвіду та знань.

Технології штучного інтелекту можуть забезпечити більш точне та швидке передбачення цін на нерухомість. Для цього можуть використовуватися різні методи машинного навчання, такі як нейронні мережі, алгоритми класифікації та кластеризації, регресійна аналітика та інші. Також для передбачення цін можуть використовуватися аналітичні моделі, які враховують різні фактори, що впливають на ціну нерухомості.

Одним з основних переваг використання штучного інтелекту у розподілених ріелторських системах є збільшення ефективності процесу передбачення цін на житло. Завдяки використанню штучного інтелекту можна автоматизувати процес збору та аналізу великої кількості даних про нерухомість, що дозволяє швидше та точніше встановлювати ціни на нерухомість. Також, за допомогою штучного інтелекту можна враховувати різні фактори, що впливають на ціну, які можуть бути складні для аналізу експертами.

Крім того, використання штучного інтелекту у розподілених ріелторських системах дозволяє зменшити витрати на оцінку нерухомості, оскільки процес

передбачення цін стає більш автоматизованим та швидким. Це може дозволити знизити вартість послуг ріелторів та забезпечити більш доступний доступ до ринку нерухомості для покупців та продавців.

Проте, використання штучного інтелекту у розподілених ріелторських системах також може мати свої недоліки. Один з них полягає у тому, що алгоритми машинного навчання можуть бути неправильно налаштовані, що може призвести до неточних результатів передбачення цін. Також можуть виникнути етичні проблеми, пов'язані з використанням даних користувачів для передбачення цін на нерухомість.

У цілому, аналіз предметної області передбачення цін житла засобами штучного інтелекту у розподілених ріелторських системах показує, що використання цих технологій може забезпечити більш точні та швидкі результати встановлення цін на нерухомість, що може мати позитивний вплив на ринок нерухомості. Однак, необхідно також звернути увагу на можливі недоліки та ризики, пов'язані з використанням штучного інтелекту у цій галузі, та розглянути способи їх запобігання.

Для запобігання неточностей та неправильного налаштування алгоритмів машинного навчання можуть використовуватись методи верифікації результатів, тестування та налаштування алгоритмів. Також необхідно забезпечити відповідну якість та об'єм даних, на яких базується передбачення цін на житло.

Щодо етичних питань, пов'язаних з використанням даних користувачів, можна використовувати методи анонімізації даних та дотримання правил конфіденційності. Також можуть бути встановлені етичні принципи, що повинні бути дотримані під час використання штучного інтелекту у розподілених ріелторських системах.

Отже, аналіз предметної області передбачення цін житла засобами штучного інтелекту у розподілених ріелторських системах дозволяє зрозуміти переваги та недоліки використання цих технологій у цій галузі. Використання штучного інтелекту може покращити ефективність та точність передбачення цін на житло, що може мати позитивний вплив на ринок нерухомості. Однак,

необхідно також враховувати ризики та недоліки, пов'язані з використанням штучного інтелекту, та використовувати методи запобігання цим проблемам.

1.2 Аналіз сучасних способів передбачення ціни житла у ріелторських системах

При аналізі сучасних способів передбачення, ми розглянемо усі аспекти і охопимо всі сфери, де застосовується передбачення, щоб обрати одну із систем та довести чому наш вибір найкращий. Сучасні способи передбачення умовно можна поділити наступним чином:

Статистичні моделі: ці моделі засновані на статистичних методах та алгоритмах, таких як регресія, класифікація та кластеризація. Вони використовують статистичні методи для аналізу даних та визначення зв'язків між різними змінними.

Математичні моделі: ці моделі використовують математичні формули та рівняння для визначення зв'язків між різними змінними. Ці моделі можуть бути досить складними та потребувати великої кількості даних для їх побудови та використання.

Експертні системи: ці системи використовують знання та експертизу людей для вирішення проблем. Вони можуть використовувати різні методи та алгоритми, щоб забезпечити якісне передбачення.

Інформаційні системи: ці системи використовують великі обсяги даних та аналітичні методи для передбачення майбутніх подій та трендів. Вони можуть використовувати різні джерела даних, такі як соціальні мережі та бази даних, для аналізу даних та визначення зв'язків між ними.

Системи прогнозування: ці системи використовують статистику та інші методи для прогнозування майбутніх подій та трендів. Вони можуть використовувати різні методи, такі як аналіз часових рядів та регресійний аналіз, для передбачення майбутніх подій.

Моделювання систем: ці системи використовують математичні моделі та алгоритми для відтворення функціонування різних систем. Наприклад, можуть

бути створені моделі для передбачення погоди, поведінки фінансових ринків або руху транспорту.

Системи аналізу даних: ці системи використовують методи статистичного аналізу та машинного навчання для визначення зв'язків між різними змінними та передбачення майбутніх подій.

Економетричні моделі: ці моделі використовують методи економічного аналізу та статистики для визначення зв'язків між різними економічними змінними та передбачення майбутніх трендів у галузі економіки.

Моделювання на основі системних даних: ці системи використовують великі обсяги даних про різні системи, такі як комп'ютерні мережі або біологічні системи, для побудови моделей та передбачення їх функціонування.

Системи розпізнавання образів: ці системи використовують методи розпізнавання образів для класифікації даних та передбачення майбутніх подій. На-приклад, вони можуть бути використані для передбачення того, які товари будуть куплені клієнтами у майбутньому на основі їх попередніх покупок.

Проте, найкращими є системи штучного інтелекту. Системи штучного інтелекту, які базуються на машинному навчанні та глибокому навчанні, є особливо корисними для задач передбачення. Ці системи аналізують великі обсяги даних та виявляють складні зв'язки між різними змінними, що дозволяє зробити точні передбачення. Зокрема, системи нейронних мереж можуть здійснювати передбачення в реальному часі, швидко адаптуватись до нових даних та враховувати багатофакторні впливи на результат. Більшість завдань передбачення вимагають швидкого та точного рішення, що робить системи штучного інтелекту найбільш ефективними інструментами для таких задач.

1.3 Аналіз методів та засобів штучного інтелекту в задачах передбачення

Розділ «Аналіз методів та засобів штучного інтелекту в задачах передбачення» включає у себе огляд найбільш популярних технологій та методів

передбачення засобами штучного інтелекту, у нашому випадку передбачення цін на житло у ріелторських системах, що використовуються сьогодні.

Один з найпоширеніших методів - регресійна модель (рис. 1.1), яка базується на аналізі залежності між ціною на житло та характеристиками нерухомості. Цей метод дозволяє визначити, як впливають певні фактори (наприклад, площа квартири, кількість кімнат, розташування, стан будинку тощо) на ціну на житло. Результатом регресійної моделі є формула, яка може використовуватись для передбачення ціни на житло на основі введених характеристик.

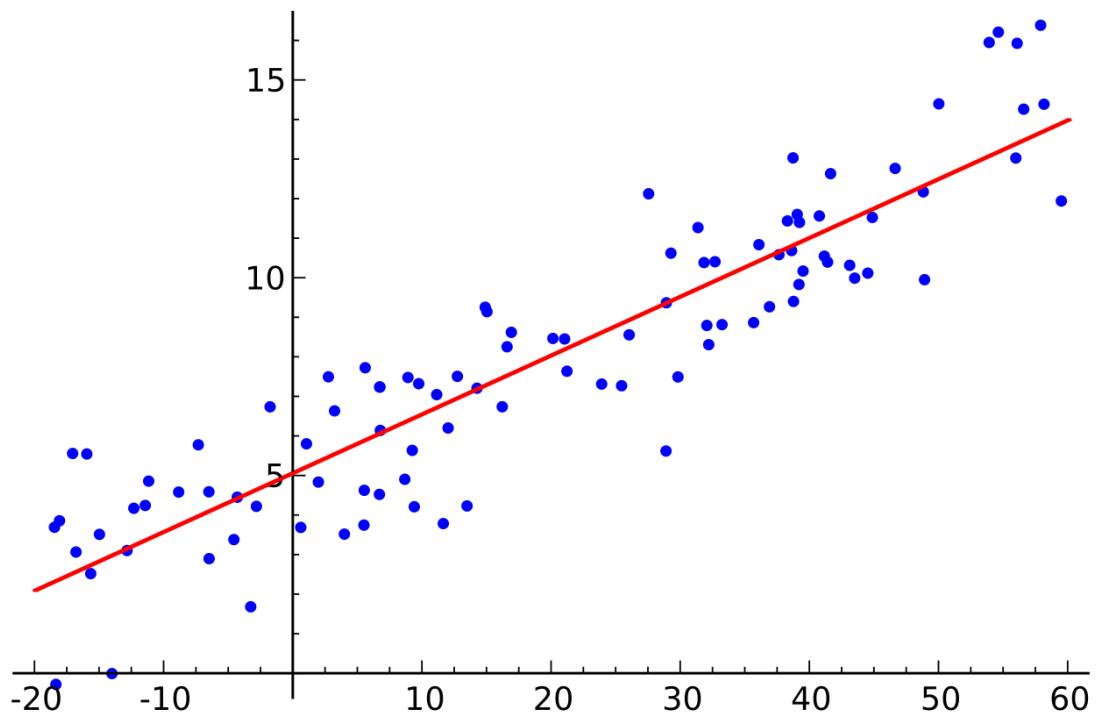


Рисунок 1.1 – Графік лінійної регресії випадкових даних

Інший метод - дерево рішень (рис. 1.2), яке використовується для аналізу великої кількості даних та виявлення залежностей між різними факторами та цінами на житло. Цей метод дозволяє розбити велику базу даних на менші підгрупи, що сприяє точнішому передбаченню цін на житло для конкретних підгруп.

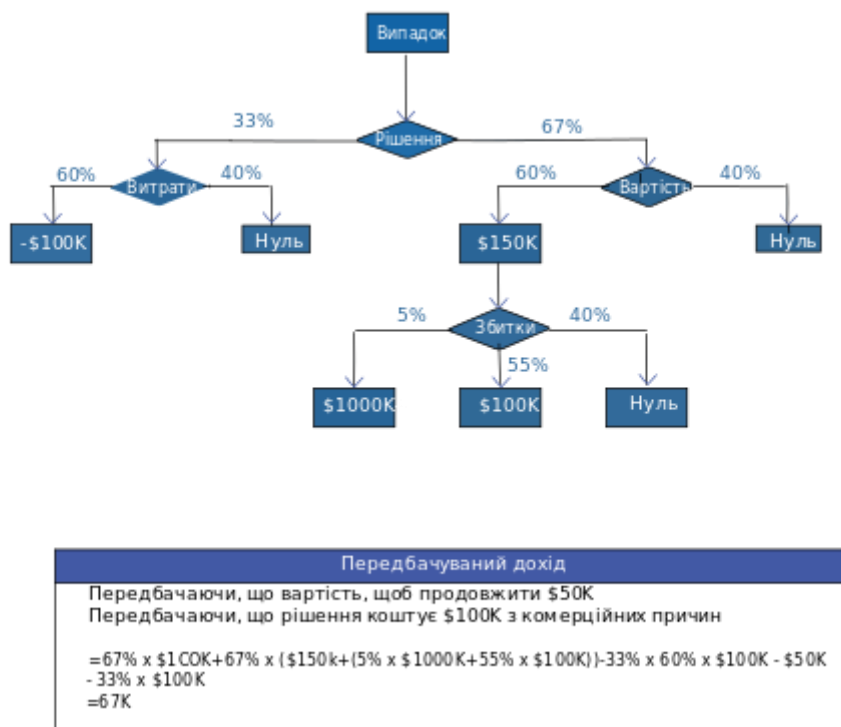


Рисунок 1.2 – Один із прикладів дерева рішень

Ще один популярний метод - нейронні мережі (рис 1.3), які використовуються для передбачення цін на житло на основі великої кількості даних. Нейронні мережі складаються зі штучних нейронів, які обробляють вхідні дані та генерують вихідні дані, які є передбаченнями цін на житло.

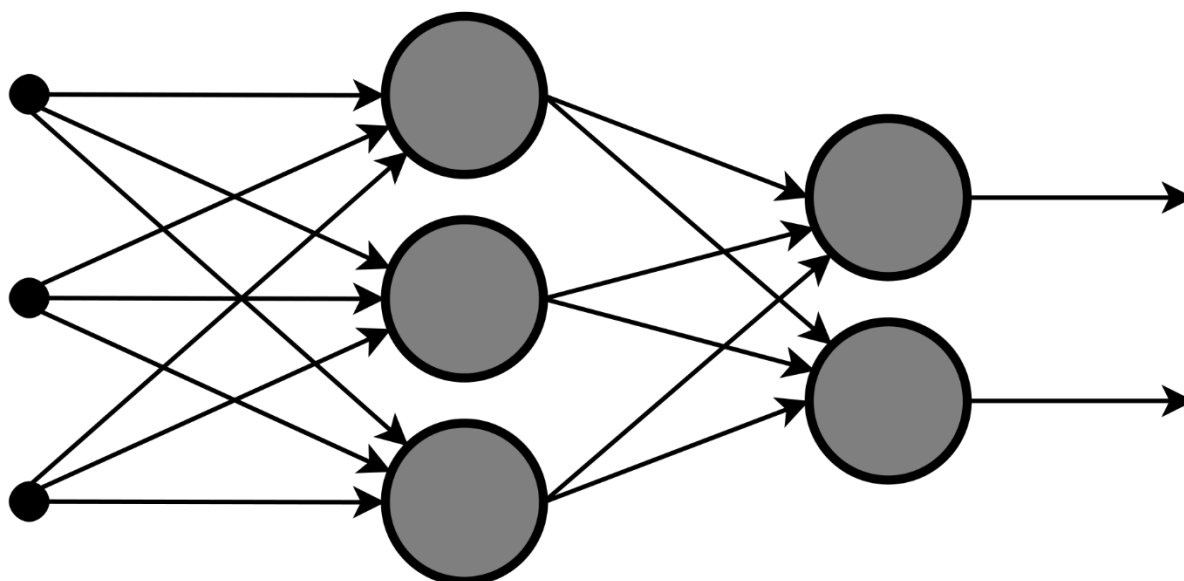


Рисунок 1.3 – Модель нейронної мережі із кількома шарами

Також використовуються генетичні алгоритми (рис. 1.4), які можуть допомогти в оптимізації параметрів моделі та поліпшенні точності передбачення цін на житло.

Генетичні алгоритми - це алгоритми, які використовують принципи еволюції для вирішення задач машинного навчання. Генетичний алгоритм використовує випадковість та відбір для знаходження оптимального рішення. У випадку передбачення цін на житло, генетичні алгоритми можуть використовуватися для знаходження оптимальних комбінацій характеристик нерухомості та їх вагових коефіцієнтів, які найкраще пояснюють залежність між характеристиками та ціна-ми на житло.

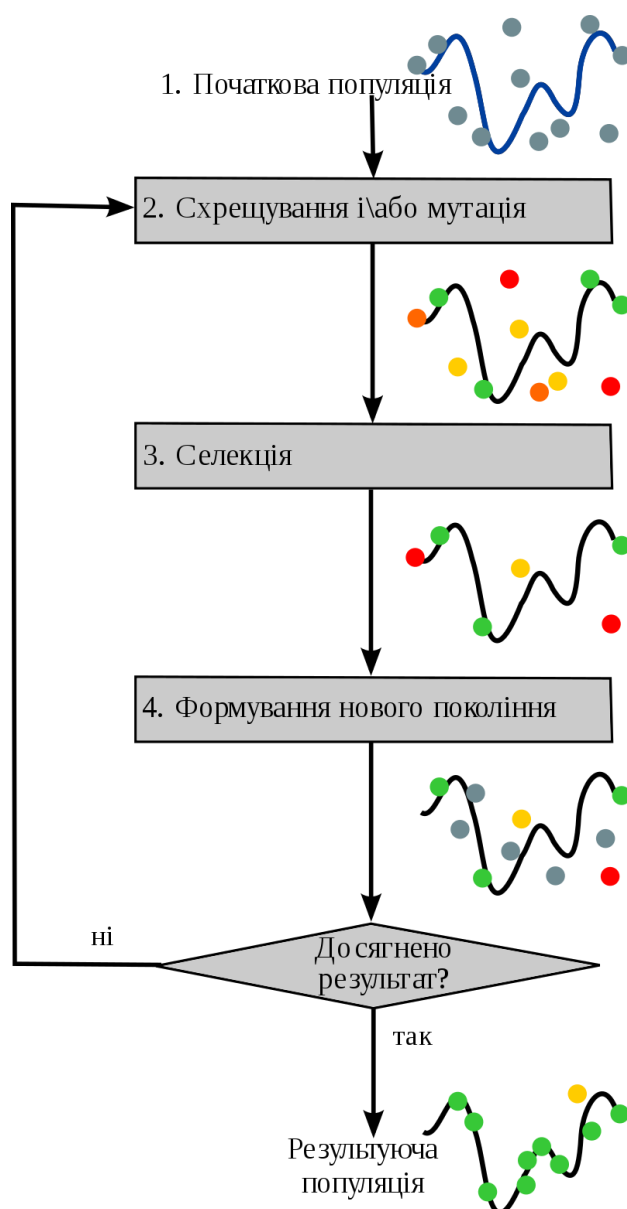


Рисунок 1.4 – Схема роботи генетичного алгоритму

Проте найкращий варіант для вирішення задач із передбачення це нейронні мережі, які можуть використовуватися для передбачення цін на житло на основі великої кількості даних. Нейронні мережі складаються зі штучних нейронів, які обробляють вхідні дані та генерують вихідні дані, які є передбаченнями цін на житло.

В цілому, передбачення цін на житло засобами штучного інтелекту у розподілених ріелторських системах є перспективним напрямком розвитку. Застосування різних методів та аналіз їх ефективності дозволить досягти більш точних результатів та поліпшити роботу ріелторських систем. Однак, важливо пам'ятати про необхідність постійного вдосконалення та оптимізації методів передбачення цін на житло, щоб забезпечити найвищу точність та якість обслуговування клієнтів.

1.4 Аналіз існуючих рішень для розподілених систем передбачення ціни житла в ріелторських установах

Розподілені системи передбачення ціни житла в ріелторських установах використовуються для збору та обробки даних з різних джерел та їх подальшого аналізу. Це дозволяє більш точно передбачати ціну житла та забезпечувати краще обслуговування клієнтів. У цьому розділі буде проаналізовано існуючі рішення для розподілених систем передбачення ціни житла в ріелторських установах.

Одним з таких рішень є система Zillow [11], яка використовує дані про ціни житла з різних джерел, таких як оголошення про продаж, об'яви про здачу в оренду та інші, щоб зробити прогнози цін на майбутнє. За допомогою глибоких нейронних мереж, система аналізує дані та робить прогнози на основі багатьох факторів, таких як розмір житла, розташування, стан будинку та інші. Відомо, що система Zillow досить точно передбачає ціну житла, але проблема полягає в тому, що вона працює тільки в США та Канаді.

Іншим рішенням є система Redfin [8], яка використовує алгоритми машинного навчання для передбачення ціни житла в різних регіонах. Система

збирає дані з різних джерел, таких як місцеві оголошення про продаж та здачу в оренду, а також дані з Інтернету та соціальних мереж. За допомогою генетичних алгоритмів та дерев рішень, система навчається передбачати ціну житла в різних регіонах. Redfin також надає користувачам можливість використовувати їхні алгоритми для аналізу різних факторів, що впливають на ціну житла, таких як шкільні райони, громадський транспорт та інші.

Ще одним прикладом розподіленої системи передбачення ціни житла є Trulia [10], яка використовує алгоритми машинного навчання та статистичні методи для передбачення цін на житло. Trulia збирає дані з різних джерел, таких як ріелторські бази даних, оголошення про продаж та здачу в оренду, а також дані про забудовників та ринок нерухомості. За допомогою аналізу даних, система навчається передбачати ціни на житло в різних місцях та у різний час.

У кінці 2020 року вийшла нова розподілена система передбачення ціни житла – HouseCanary[3], яка використовує аналітичні та статистичні методи для збору та обробки даних. HouseCanary працює з даними про продажі та оренду житла, оцінки вартості житла та різних економічних факторів. За допомогою аналізу даних та машинного навчання, система навчається передбачати ціни житла в різних місцях та на різний час.

Україна також має свої власні рішення для передбачення цін на житло. Одним з них є портал Dom.ria [1], який є одним з найбільших порталів нерухомості в Україні. Dom.ria використовує аналіз даних, щоб допомогти користувачам знайти найбільш точну ціну на житло. Портал збирає дані з різних джерел, таких як оголошення про продаж та здачу в оренду, а також дані з місцевих агентств нерухомості.

Іншим українським рішенням для передбачення цін на житло є портал нерухомості Lun.ua [5]. Вони використовують аналіз даних та машинне навчання для надання користувачам точної інформації про ціни на житло в Україні.

Lun.ua збирає дані з різних джерел, таких як оголошення про продаж та здачу в оренду, а також з місцевих агентств нерухомості. Вони використовують ці дані, щоб розробити свої алгоритми машинного навчання та аналізу даних, які допомагають передбачати ціни на житло. Крім того, Lun.ua проводить аналіз

ринку нерухомості та враховує такі фактори, як розташування житла, тип житла, розмір та інші характеристики, щоб забезпечити точність передбачення цін на житло.

Ще одним українським рішенням є система Price.ua [7], яка не тільки надає інформацію про ціни на житло, але і порівнює ціни на різні товари та послуги в Україні. Price.ua збирає дані з різних джерел, включаючи оголошення про продаж та здачу в оренду, а також інформацію про ціни в різних магазинах та інтернет-магазинах. За допомогою своїх алгоритмів машинного навчання та аналізу даних, Price.ua навчається передбачати ціни на різні товари та послуги, включаючи ціни на житло.

Українські рішення для передбачення цін на житло використовують різні технології та джерела даних для забезпечення найбільш точної інформації. Ці рішення допомагають користувачам знайти найбільш вигідні пропозиції на ринку нерухомості в Україні, а також зробити обґрунтований вибір при покупці або здачі в оренду житла.

У загальному, існує багато різних розподілених систем передбачення ціни житла в ріелторських установах, кожна з яких має свої переваги та недоліки. Розподілені системи дозволяють збирати та аналізувати більше даних для більш точного передбачення цін на житло та забезпечення кращого обслуговування клієнтів.

1.5 Мета, задачі та вимоги до реалізації способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

Мета кваліфікаційної роботи полягає в створенні способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

Для досягнення поставленої мети визначені наступні задачі дослідження:

– аналіз сучасних способів передбачення ціни житла ріелторських системах;

– аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем;

- реалізувати інформаційну технологію способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах;
- провести експериментальне тестування інформаційної технології.

Розділ 2 Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

2.1 Загальна послідовність задачі передбачення

Послідовність передбачення здійснює прогноз на основі даних, взятих на основі певного часу спостереження, взаємодія із моделлю прогнозу, підбір оптимальних параметрів для передбачення. У результаті, на основі цих даних робиться передбачення (рис. 2.1).

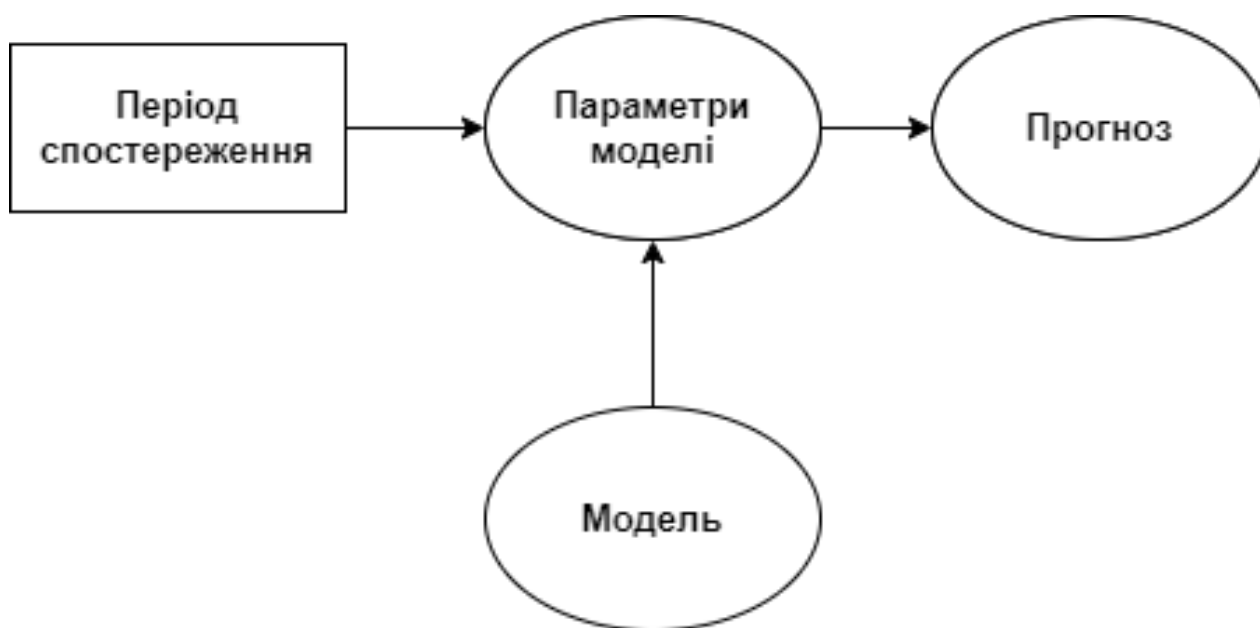


Рисунок 2.1 – Схема послідовності передбачення

При вирішенні задач передбачення із використанням складних, багатокритеріальних даних потрібно чітко визначити оптимальні параметри моделі. Для цього застосовується штучний інтелект. Застосування штучного інтелекту у процесі передбачення дозволяє оптимізувати вибір параметрів моделі. ШІ може проводити велику кількість ітерацій, перебираючи різні комбінації параметрів, щоб знайти найкращі значення. Такий підхід надає можливість збільшити точність та надійність прогнозу. Використання штучного інтелекту допоможе виявити складні залежності між даними та визначити оптимальні сполучення параметрів, що призводять до кращих передбачень. (рис. 2.2).



Рисунок 2.2 – Схема послідовності передбачення із застосуванням ШІ

2.2 Загальна постановка задачі передбачення ціни житла

Загальна постановка задачі передбачення за допомогою способів штучного інтелекту, а саме для передбачення ціни житла сформована так:

На основі наявних даних про житло та попередньо проаналізованих характеристик штучного інтелекту, реалізувати передбачення ціни житла, що буде відповідати заданим умовам та вимогам.

Дані про житло передбачають охоплення характеристик, що впливають на ціну житла, а тобто:

1. Характеристики, що відображають загальний вигляд житла
 - Оцінка житла
 - Район
2. Характеристики, що відображають розташування житла
 - Довгота
 - Широта
3. Характеристики, що відображають додаткову інформацію про житло

- Рік побудови житла
- Рік проведення ремонту
- Поштовий індекс

4. Характеристики, що відображають наявну інфраструктуру

- Розташування соціальних закладів, магазинів тощо
- Наявність набережної

5. Характеристики площі житла

- Загальна площа
- Площа проживання
- Площа підвалу
- Площа проживання (без підвалу)
- Площа до/після ремонту

Класифікація факторів та даних про житло за певними чинниками полегшує їх аналіз та спрощує розуміння їх впливу на ціну житла загалом. Попередня класифікація характеристик, що впливають на кінцеву ціну житла та повинні враховуватись при виборі вибірки даних зображені за допомогою схеми на рисунку 2.3.

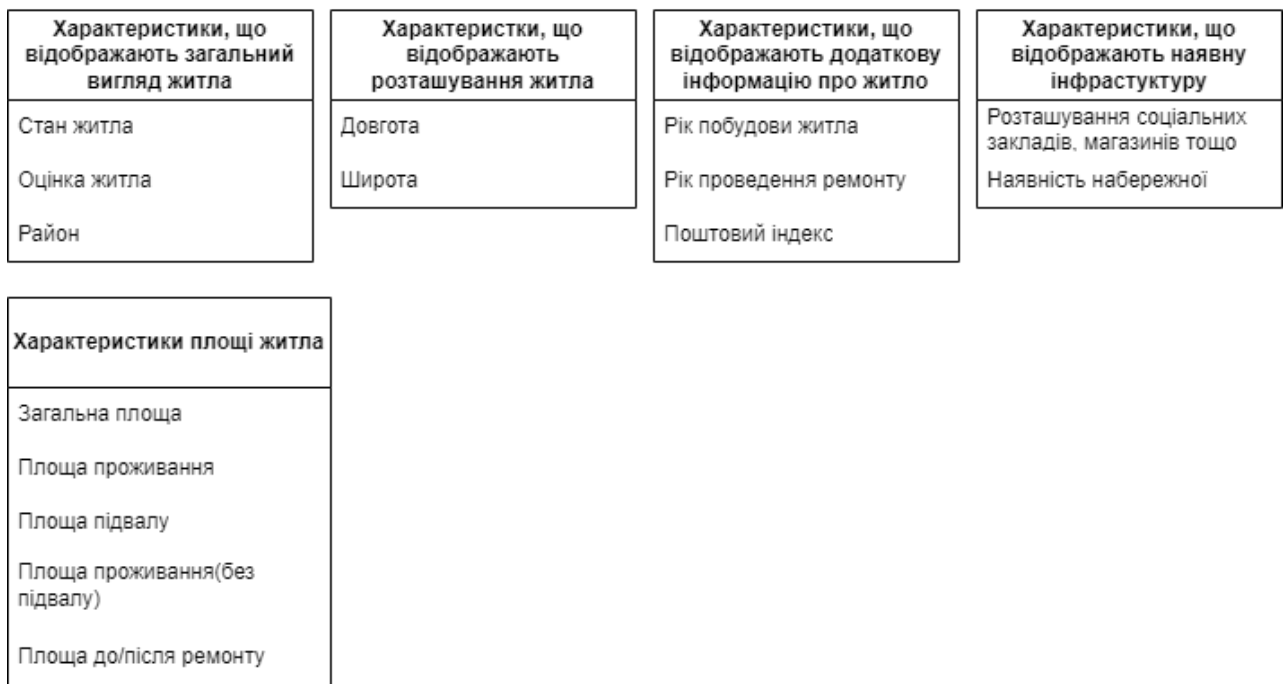


Рисунок 2.3 – Класифікація характеристик житла, що впливають на кінцеву ціну

Вимоги до штучного інтелекту (рис. 2.4):

- Низька похибка передбачення
- Точність передбачення $> 70\%$

Кінцевий результат повинен відповідати усім вимогам та мати можливість до використання набутого досвіду у подальших задачах із передбачення, мати чітку та сформовану взаємодію із кінцевим користувачем.

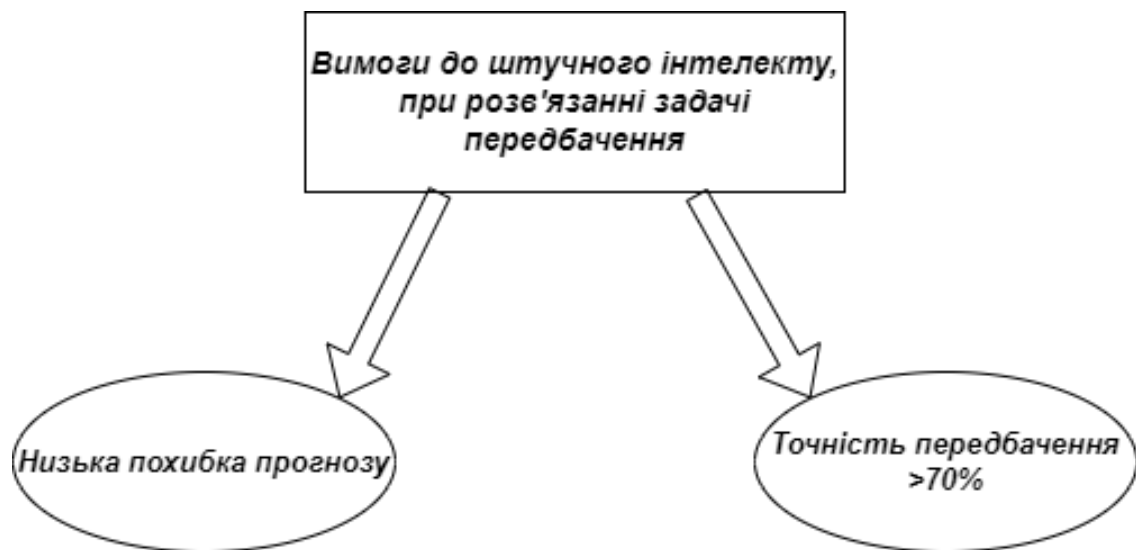


Рисунок 2.4 – Вимоги до штучного інтелекту

2.3 Загальна послідовність передбачення ціни житла

Послідовність передбачень за допомогою методів штучного інтелекту включає наступні кроки:

- Підготовка даних
- Тренування моделі
- Валідація моделі
- Налаштування гіперпараметрів
- Передбачення.

На рисунку 2.5 зображено схему послідовності вирішення задачі із передбачення штучним інтелектом.



Рисунок 2.5 – Загальна схема послідовності передбачення штучним інтелектом

Підготовка даних є критичним кроком у створенні моделі штучного інтелекту, оскільки якість вхідних даних суттєво впливає на продуктивність моделі. Процес підготовки даних включає такі кроки:

- Очищення даних: Цей крок передбачає виявлення та обробку відсутніх значень, неправильних значень та інших видів помилок в даних.

- Трансформація даних: Цей крок передбачає перетворення даних у формат, який буде використаний алгоритмом штучного інтелекту. Дані містять категоріальні змінні перетворюються у числові змінні перед використанням алгоритму. Цей крок включає масштабування та нормалізацію даних, а також створення нових ознак з наявних даних.

- Розбиття даних: Після очищення та трансформації дані поділяються на тренувальний та тестовий набори даних. Тренувальний набір даних використовується для навчання моделі штучного інтелекту, тоді як тестовий набір даних використовується для оцінки продуктивності моделі на валідаційній вибірці.

Навчання моделі – це процес налаштування параметрів моделі штучного інтелекту за допомогою набору даних для тренування з метою мінімізації різниці між передбаченим виводом і фактичним виводом. Цей процес виконується за

допомогою алгоритму оптимізації, що налаштовує параметри моделі на основі помилки між передбаченим виводом і фактичним виводом.

Процес тренування розбитий на наступні кроки:

– Ініціалізація: Перший крок у тренуванні моделі – це ініціалізація параметрів моделі. Це робиться випадково або за допомогою попередньо навченої моделі, підбираючи оптимальні параметри.

– Тренування: Наступний крок – використання ініціалізованої моделі для здійснення передбачень на тренувальній вибірці. Це включає передачу вхідних даних через шари моделі ШІ для отримання виводу.

– Обчислення похибки: Після того, як модель пройшла тренування, різниця між передбаченим виводом і фактичним виводом обчислюється за допомогою функції похибки. Функція похибки вимірює помилку між передбаченим виводом і фактичним виводом.

– Тренування на основі обчисленої : Обчислена функцією похибки помилка використовується для коригування параметрів моделі. Це робиться за допомогою алгоритму зворотного поширення помилки.

– Повторення: Процес повторюється протягом кількох ітерацій або епох до того моменту, коли продуктивність моделі на тренувальних даних буде оптимізована.

Перевірка моделі – це процес оцінки продуктивності моделі штучного інтелекту на окремому наборі даних, який не використовувався для тренування. Метою перевірки моделі є переконання, що модель може узагальнюватися на нові, невидимі дані та не перенавчається на тренувальних даних. Перевірка передбачає валідацію моделі ШІ, тобто модель ШІ оцінюється за допомогою набору даних для валідації, щоб переконатися, що вона може узагальнюватися на нові дані.

Гіперпараметри – це параметри моделі штучного інтелекту, які встановлюються до початку процесу тренування і не можуть бути вивчені з даних. Прикладами гіперпараметрів є швидкість навчання, розмір вибірки до ітерації, кількість епох, кількість прихованих шарів та кількість нейронів у кожному шарі тощо.

Налаштування гіперпараметрів – це процес вибору найкращих гіперпараметрів для моделі штучного інтелекту з метою оптимізації її продуктивності при виконанні певного завдання. Налаштування гіперпараметрів важливе, оскільки продуктивність моделі штучного інтелекту може сильно залежати від вибору гіперпараметрів. Непідходящі гіперпараметри можуть призвести до поганої продуктивності, включаючи недостатнього натренування або перенавантаження моделі ШІ.

Коли модель штучного інтелекту навчена та перевірена, вона використовується для передбачення результатів на нових даних. Процес передбачення включає наступні кроки:

- Попередня обробка даних: Вхідні дані повинні бути попередньо оброблені так само, як тренувальна та валідаційна вибірки, щоб модель могла правильно інтерпретувати дані.

- Пряма передача: Вхідні дані подаються до навченої моделі.

- Форматування результату: Нарешті, результат передбачення форматується таким чином, щоб він був корисним для поставленої задачі та кінцевого користувача.

2.4 Особливості застосування засобів штучного інтелекту у розподілених ріелторських системах для передбачення ціни житла

Для вирішення задачі передбачення методами штучного інтелекту використовується багатошаровий перцептрон. Це один із типів штучної нейронної мережі, що складається із декількох шарів взаємопов'язаних між собою нейронів, перцептронів. Також багатошаровий перцептрон є нейронною мережею прямого поширення, тобто інформація рухається в одному напрямку (рис 2.6).

Функцією активації багатошарового перцептрона слугує зрізаний лінійний вузол і визначається таким чином (2.1):

$$f(x) = x^+ = \max(0, x) = \begin{cases} \text{якщо } x > 0, \text{ то } x \\ 0 \end{cases} \quad (2.1)$$

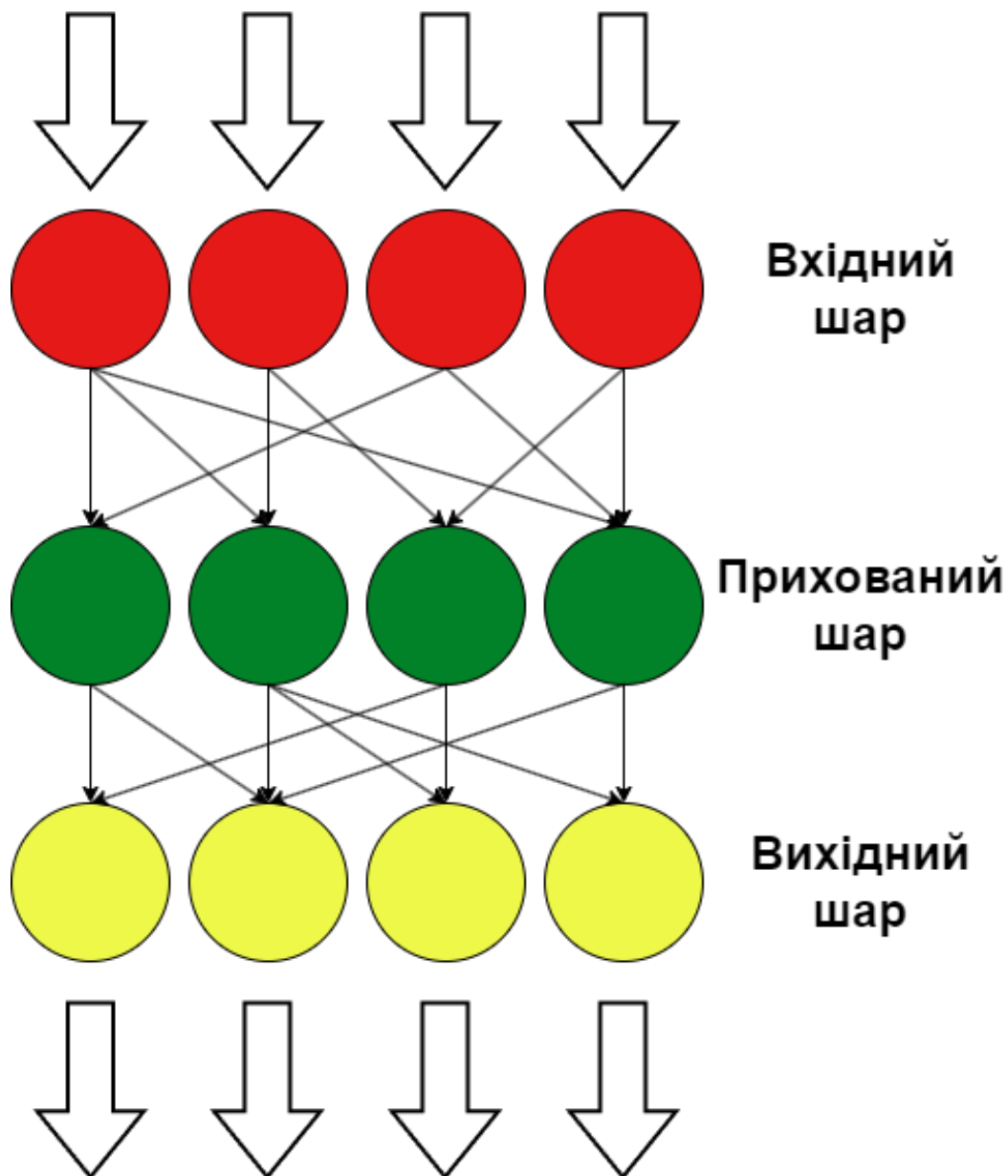


Рисунок 2.6 – Приклад нейронної мережі прямого поширення

Для перевірки правильності відпрацювання тренування моделі штучного інтелекту застосовується обрахування:

- Середньої абсолютної похибки (2.2)

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}, \quad (2.2)$$

де x_i – справжнє значення, а y_i – прогнозоване.

- Середньоквадратичної похибки (2.3)

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (2.3)$$

де Y_i – справжнє значення, а \hat{Y}_i – прогнозоване.

– Кореневе середньоквадратичне відхилення (2.4)

$$RMSE = \sqrt{\frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{T}}, \quad (2.4)$$

де \hat{y}_t – прогнозоване значення, t, T – кількість, y_t – залежна змінна регресії.

– Оцінка дисперсії випадкової величини, як метрики ризику, точності

При процесі прогнозування застосовується алгоритм зворотного поширення помилки – «backpropagation». Ця важлива складова процесу навчання нейронної мережі дозволяє вчитись даним штучним мережам на основі прогнозованих раніше ними даних, вносити корективи до параметрів, ваг нейронів і зміщень з метою мінімізації різниці між прогнозованими та фактичними даними (помилки).

Алгоритм зворотного поширення помилки, або ж «backpropagation» розраховує градієнт помилки нейронної мережі прямого поширення, враховуючи параметри нейронів та похибку.

Алгоритм зворотного поширення помилок можна розбити на наступні етапи:

– Прямий прохід: Під час прямого проходу вхідні дані передаються у нейронну мережу, і вони поширюються вперед через шари. Кожний нейрон отримує вхідні дані, застосовує активаційну функцію до зваженої суми цих вхідних даних та передає результат наступному шару. Цей процес триває до тих пір, поки вихідний шар не виробить прогнозоване значення.

– Обчислення функції втрат: Після отримання прогнозованого результату він порівнюється з фактичним значенням за допомогою функції втрат. Функція втрат вимірює розбіжність між прогнозованим та фактичним виходами та надає кількісну міру продуктивності мережі.

– Зворотній прохід: Під час зворотного проходу обчислюються градієнти функції втрат щодо ваг та зміщень мережі. Градієнти вказують, як зміниться функція втрат при невеликих змінах ваг і зміщень.

– Оновлення ваг і зміщень: Обчислені градієнти використовуються для оновлення ваг і зміщень мережі. Це оновлення виконується за допомогою алгоритму оптимізації, такого як градієнтний спуск або його варіанти. Алгоритм оптимізації коригує ваги та зміщення в напрямку, що мінімізує функцію втрат.

– Повторення: Процес прямого та зворотного проходу повторюється декілька разів або протягом кількох епох (ітерацій), щоб поліпшити продуктивність мережі. Процес прямого та зворотного проходу виконується ітеративно до досягнення задовільного рівня продуктивності мережі.

Загальну послідовність виконання алгоритму зворотної помилки зображено на рисунку 2.7.

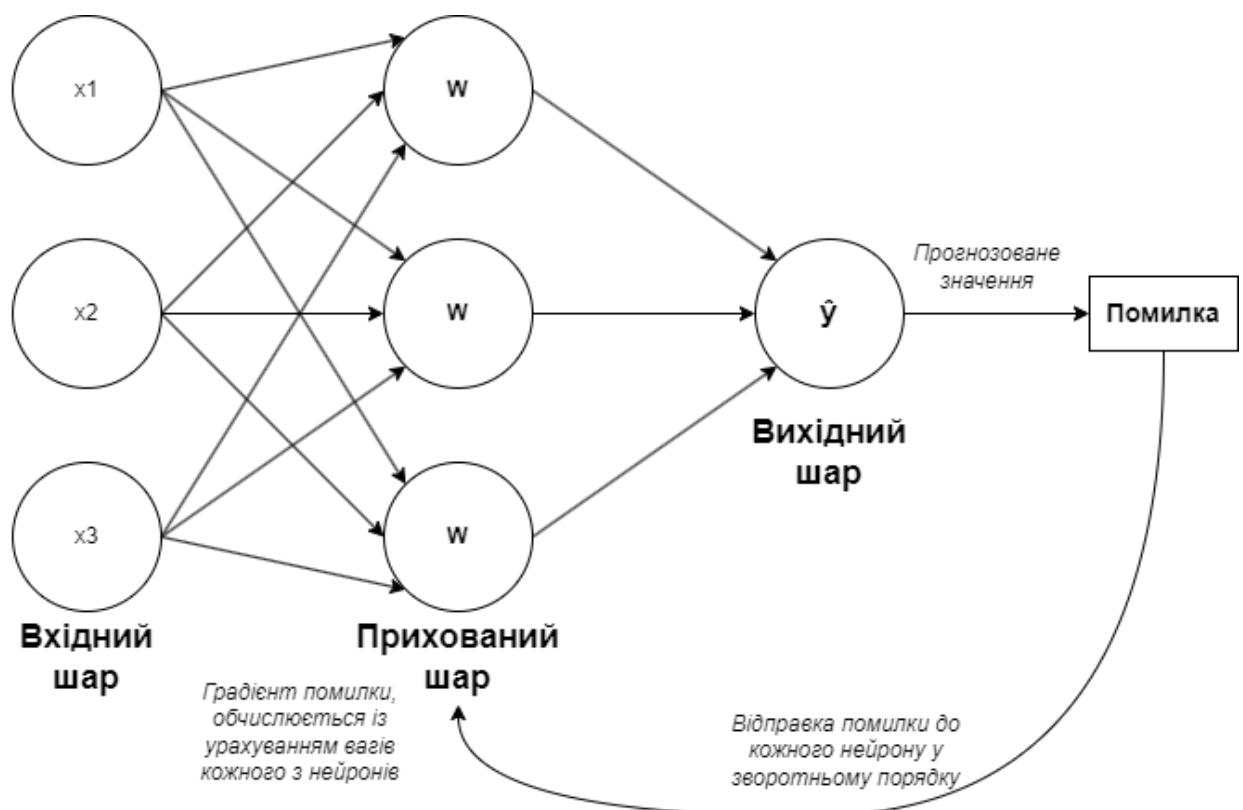


Рисунок 2.7 – Послідовність виконання алгоритму «backpropagation»

Важливо зауважити, що розроблена система повинна відповідати вимогам розподіленої ріелторської системи (PPC). Для цього буде створено

адміністративну частину, з якою можуть взаємодіяти всі користувачі, що знаходяться у даній розподіленій системі (рис. 2.8).

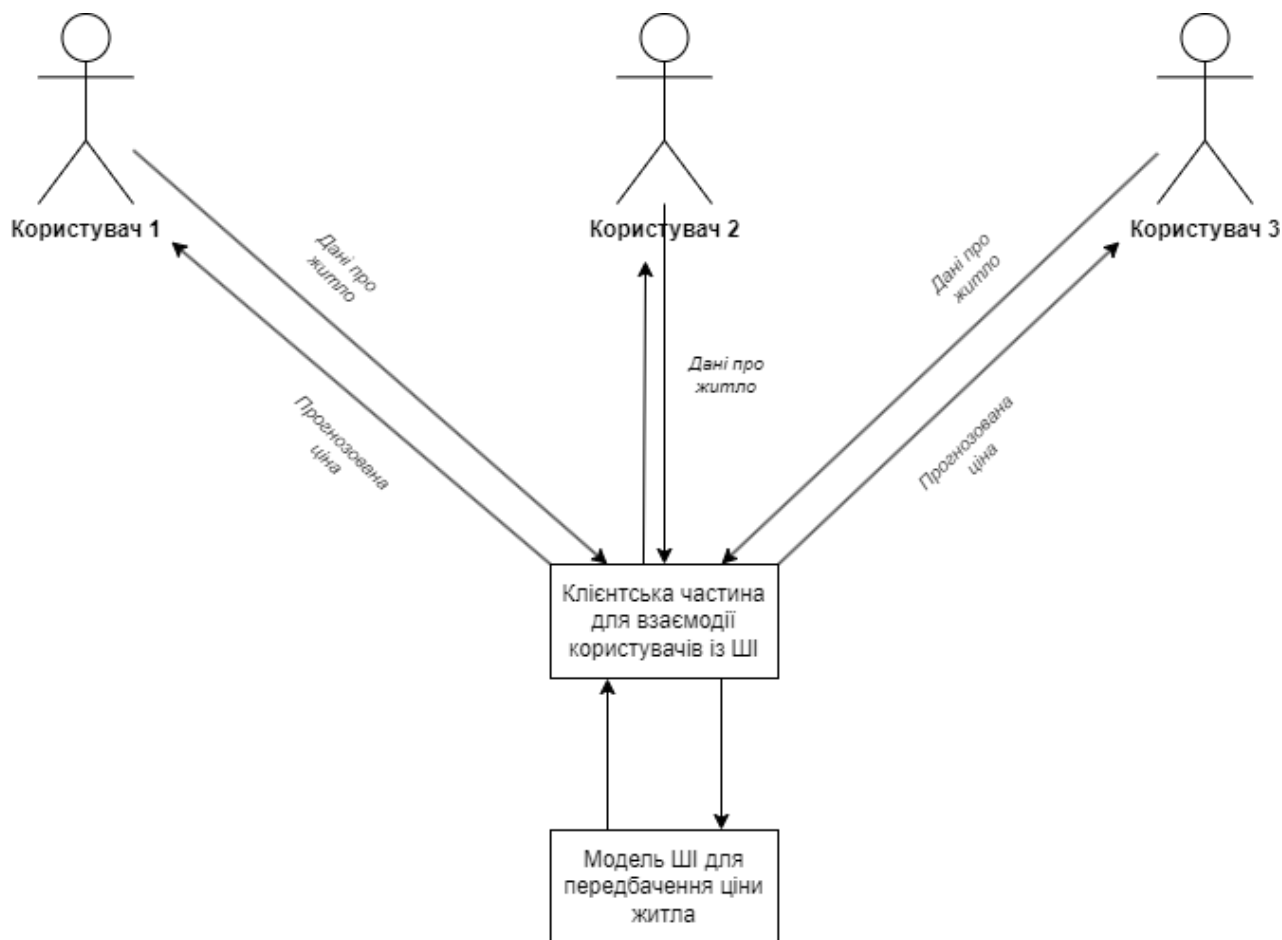


Рисунок 2.8 – Схема взаємодії користувачів РРС із моделлю ШІ

При необхідності розроблену модель штучного інтелекту для передбачення ціни житла можна застосовувати у різних незалежних середовищах, що дає змогу цілком і повністю використовувати цю модель ШІ у розподілених ріелторських системах та робити це з огляду на власні інтереси. Для взаємодії незалежних середовищ із моделлю штучного інтелекту потрібно створити проміжну ланку взаємодії цих середовищ та моделі – Application Programming Interface (API). При створенні API створюються елементи на методи взаємодії та описується їх використання. Це необхідно для того, щоб будь-хто із зацікавлених у взаємодії з моделлю прогнозування цін осіб зміг правильно налаштувати взаємодію між нею та своїм незалежним середовищем. Схему використання

розробленої моделі штучного інтелекту у розподілених ріелторських системах зображено на рисунку 2.9.

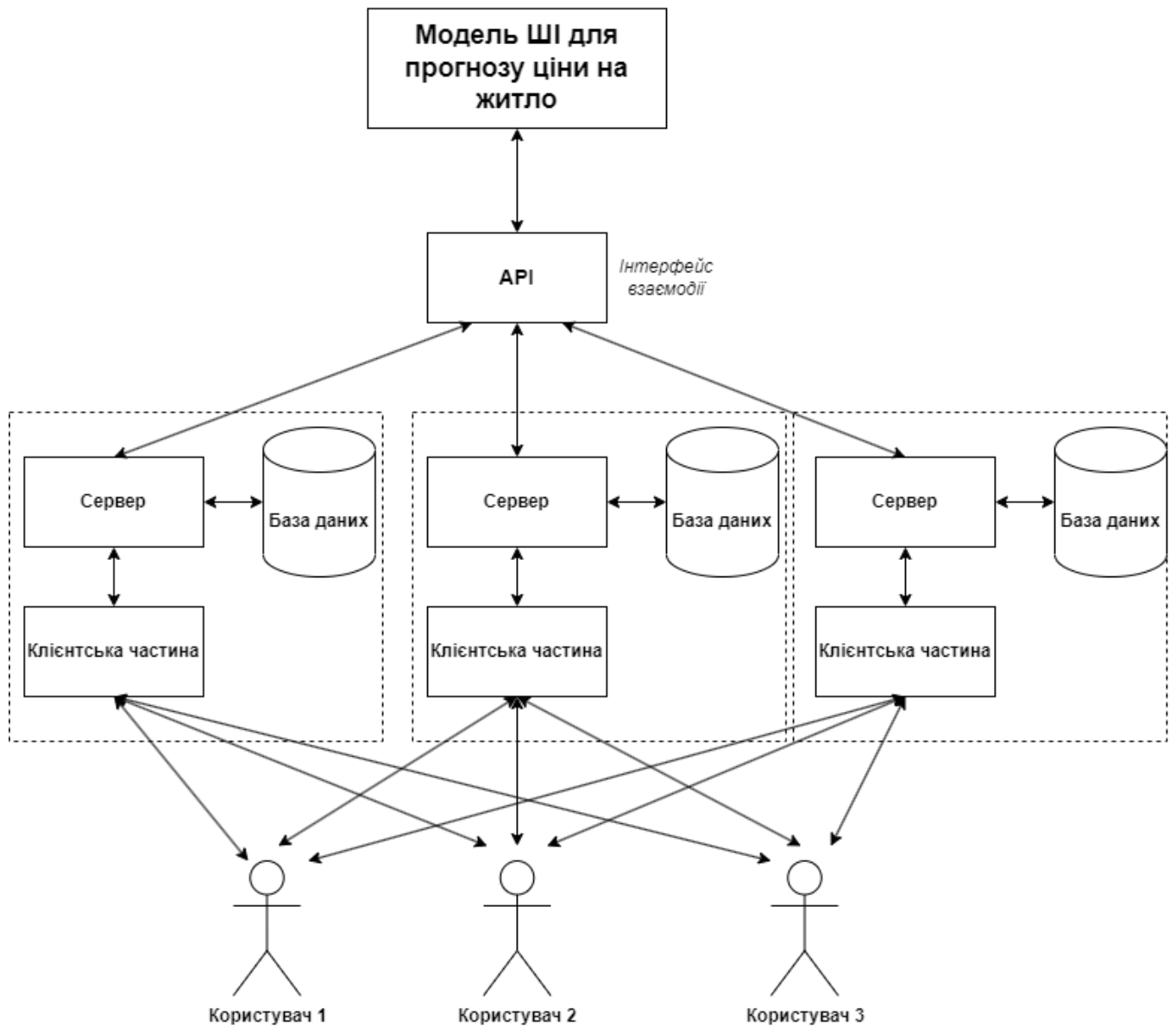


Рисунок 2.9 – Використання моделі ШІ для передбачення цін у розподілених ріелторських системах

Застосування моделі штучного інтелекту у розподілених ріелторських системах дозволяє кожній системі використовувати цю модель відповідно до своїх потреб та інтересів. Кожна система може налаштувати взаємодію з моделлю через API, щоб отримувати прогнози цін житла, необхідні для їхньої роботи та прийняття рішень.

2.5 Функціональна структура інформаційної системи

Функціональна структура інформаційної системи є важливим інструментом для розуміння та визначення функцій та процесів, які складають систему. Вона надає ієрархічне подання цих функцій та процесів і відображає зв'язки між ними. Ця структура представлена у вигляді діаграми (рис. 2.10), що складається зі зв'язків між функціями та процесами.

Реалізована діаграма демонструє необхідні функції, та їх опис для правильності виконання поставленої задачі системою в цілому.

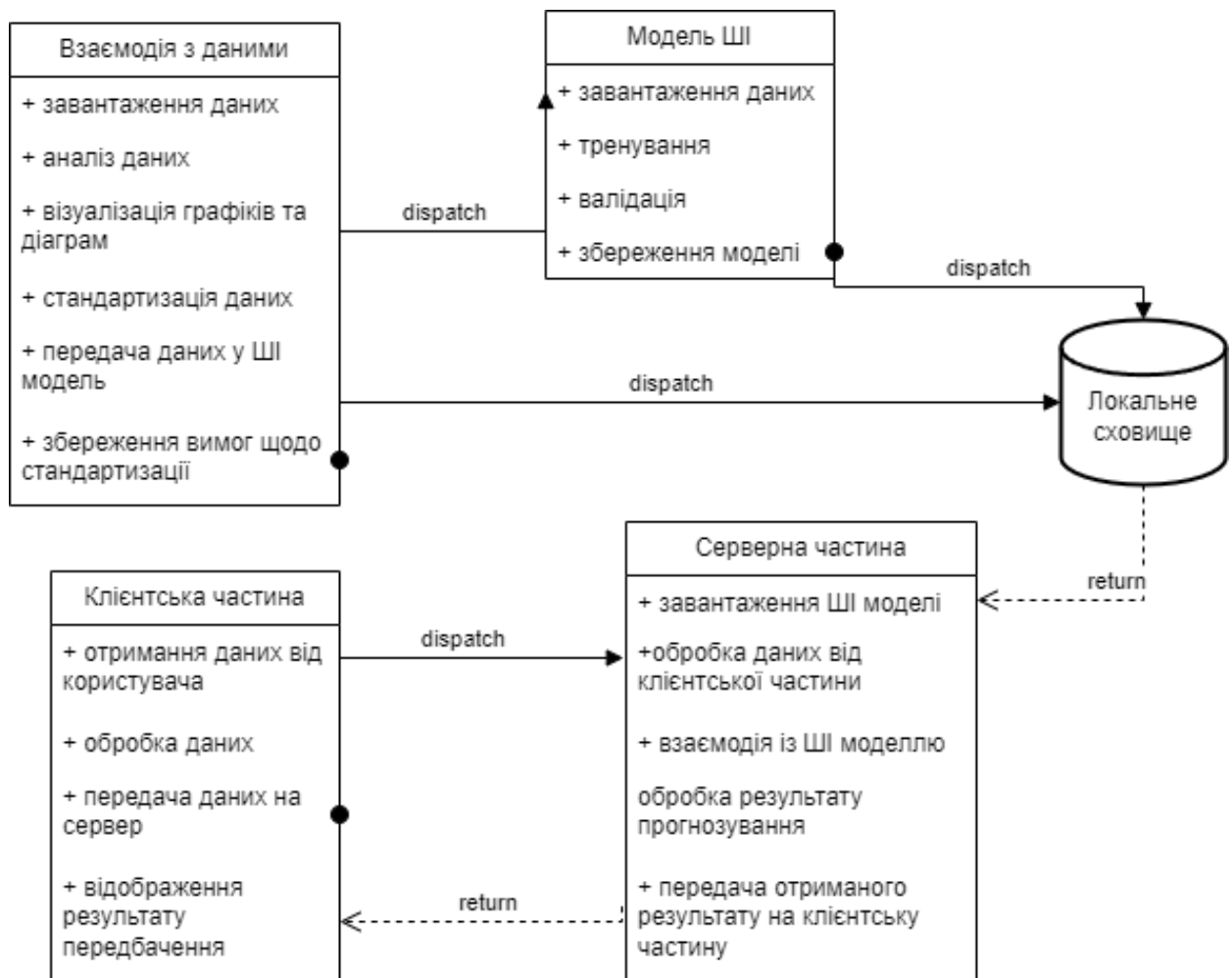


Рисунок 2.10 – Діаграма функціональної структури системи

У діаграмі функціональної структури інформаційної системи вказуються всі необхідні функції та процеси, які система повинна виконувати для успішного виконання поставлених задач. Кожна функція чи процес мають свій відповідний

опис, який деталізує їхню роль та призначення у системі. Функціональна структура інформаційної системи допомагає уникнути непорозумінь та помилок під час розробки та впровадження системи. Вона слугує основою для подальшого проектування та розробки детальних компонентів системи та їхнього взаємозв'язку.

Важливою частиною функціональної структури інформаційної системи є зв'язки між функціями та процесами. Ці зв'язки можуть вказувати на послідовність виконання функцій, взаємозалежності, передачу даних чи контрольних сигналів між функціями. Діаграма функціональної структури дозволяє легко визначити ці зв'язки та зрозуміти взаємодію між різними частинами системи.

2.6 Проектування структури інформаційної системи

На рисунку 2.11 зображено схему структури модулів системи.



Рисунок 2.11 – Структура модулів системи

Структура системи поділена на дві основні частини – серверну та клієнтську. Серверна частина є основною, що виконує процес передбачення, клієнтська у свою чергу слугує інтерфейсом користувача, що розроблений з метою досягнення найоптимальнішої, зрозумілої та ефективної взаємодії моделі ШІ та кінцевого користувача.

Серверна частина містить у собі такі модулі:

– Модуль завантаження даних: цей модуль відповідає за завантаження необхідних даних для тренування та валідації ШІ моделі. Він отримує дані із файлу у форматі «.csv».

– Модуль попереднього аналізу та візуалізації даних: цей модуль відповідає за аналіз та візуалізацію попередньо завантажених даних. Він виконує різні статистичні обчислення, виявляє кореляції між різними змінними, будує графіки та діаграми для кращого розуміння даних та покращення взаємодії із ними у подальшому.

– Модуль нормалізації даних та підготовки їх застосування: цей модуль відповідає за нормалізацію даних, а саме: масштабування змінних, видалення аномальних значень, заповнення пропущених даних. Він також готує дані для подальшого використання в ШІ моделі, змінює їхній формат та тип згідно до необхідних умов ШІ моделі.

– Модуль ініціалізації ШІ моделі: цей модуль відповідає за створення базової структури ШІ моделі. Він включає в себе вибір архітектури моделі, ініціалізацію значень, вибір оптимізатора та інших параметрів моделі.

– Модуль тренування ШІ моделі: цей модуль виконує тренування ШІ моделі на оптимізованих раніше даних. Він виконує ітерації над навчальними даними моделі, проводить попередні прогнози.

– Модуль валідації ШІ моделі: цей модуль оцінює продуктивність навченої ШІ моделі за допомогою валідаційного набору даних. Він оцінює якість прогнозів моделі, порівнюючи їх з відомими правильними відповідями. Це дозволяє виявити перенавчання або недостатню адаптацію моделі до даних.

– Модуль візуалізації результату роботи ШІ моделі: цей модуль відповідає за візуалізацію результатів прогнозування ШІ моделі. Він створює графіки, діаграми, картографічні представлення для наглядного відображення передбачених цін житла та їх залежності від різних характеристик.

Процес взаємодії модулів зображено на рисунку 2.12.



Рисунок 2.12 – Послідовність взаємодії модулів

Загалом, кожен модуль виконує конкретні завдання і взаємодіє з іншими модулями, щоб забезпечити ефективну роботу системи. Від завантаження даних до навчання ШІ моделі, від валідації результатів до обробки відповідей для користувача - кожен модуль виконує важливу функцію в процесі передбачення ціни житла. Ця структура дозволяє системі аналізувати дані, навчатися на них та надавати користувачам цінну інформацію для прийняття рішень у сфері нерухомості.

2.7 Висновки до розділу 2

Реалізовано спосіб передбачення ціни житла засобами штучного інтелекту у розподільних ріелторських системах.

Визначено загальну послідовність задачі передбачення з подальшою імплементацією методів штучного інтелекту.

Розроблено загальну послідовність передбачення ціни житла із визначенням структури собівартості та урахуванням особливостей застосування засобів штучного інтелекту у розподілених ріелторських системах.

Для вирішення задачі передбачення методами штучного інтелекту було використано багатошаровий перцептрон. Визначено фактори та чинники, що впливають на результат передбачення з використанням методів штучного інтелекту та визначено номенклатуру параметрів контролю показників передбачення, розроблено план дій до кожного із випадків.

Розроблена ріелторська система реалізована як розподілена система. Розроблено функціональну структуру системи та структурний вигляд програмних модулів. Описано та поставлено межі відповідальності кожного із модулів, блоків та елементів системи.

Розділ 3 Програмна реалізація способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

3.1 Структура модулів системи, їх взаємозв'язок

Розроблена система передбачення цін житла методами штучного інтелекту складається з різних модулів, які працюють разом для ефективного аналізу даних та прогнозування цін на нерухомість. Розроблена система поділена на два середовища – середовище створення ШІ моделі, незалежне середовище використання готової натренованої моделі. На рисунку 3.1 зображено схему структури модулів цієї системи.

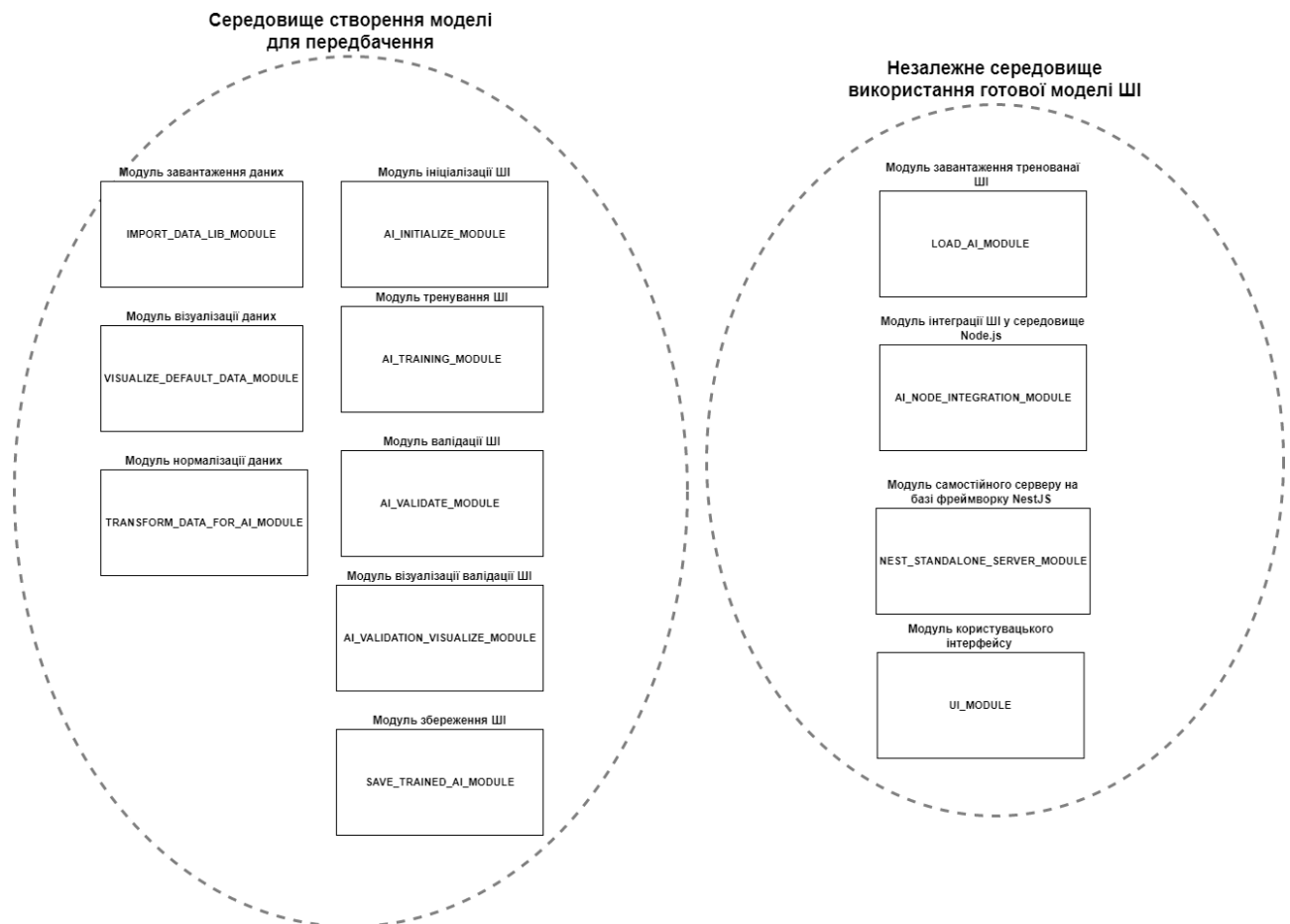


Рисунок 3.1 – Структура програмних модулів системи

Ці два середовища не взаємодіють напряму, натомість вони використовують локальне сховище для збереження ШІ та завантаження тренованої ШІ відповідно (рис. 3.2).

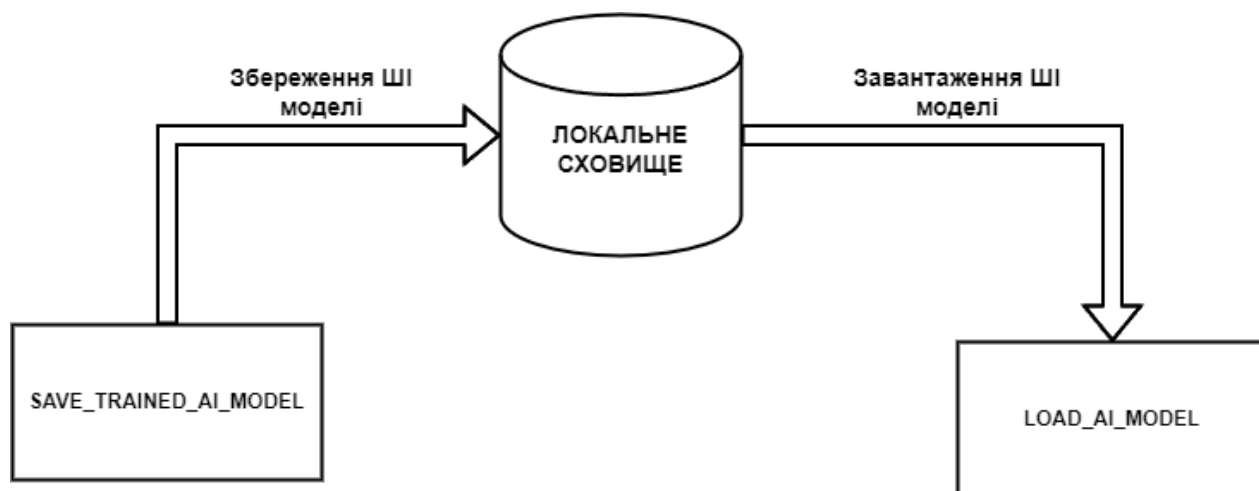


Рисунок 3.2 – Принцип взаємодії двох середовищ

Взаємодію програмних модулів у середовищі створення ШІ моделі схематично відображає рисунок 3.3.

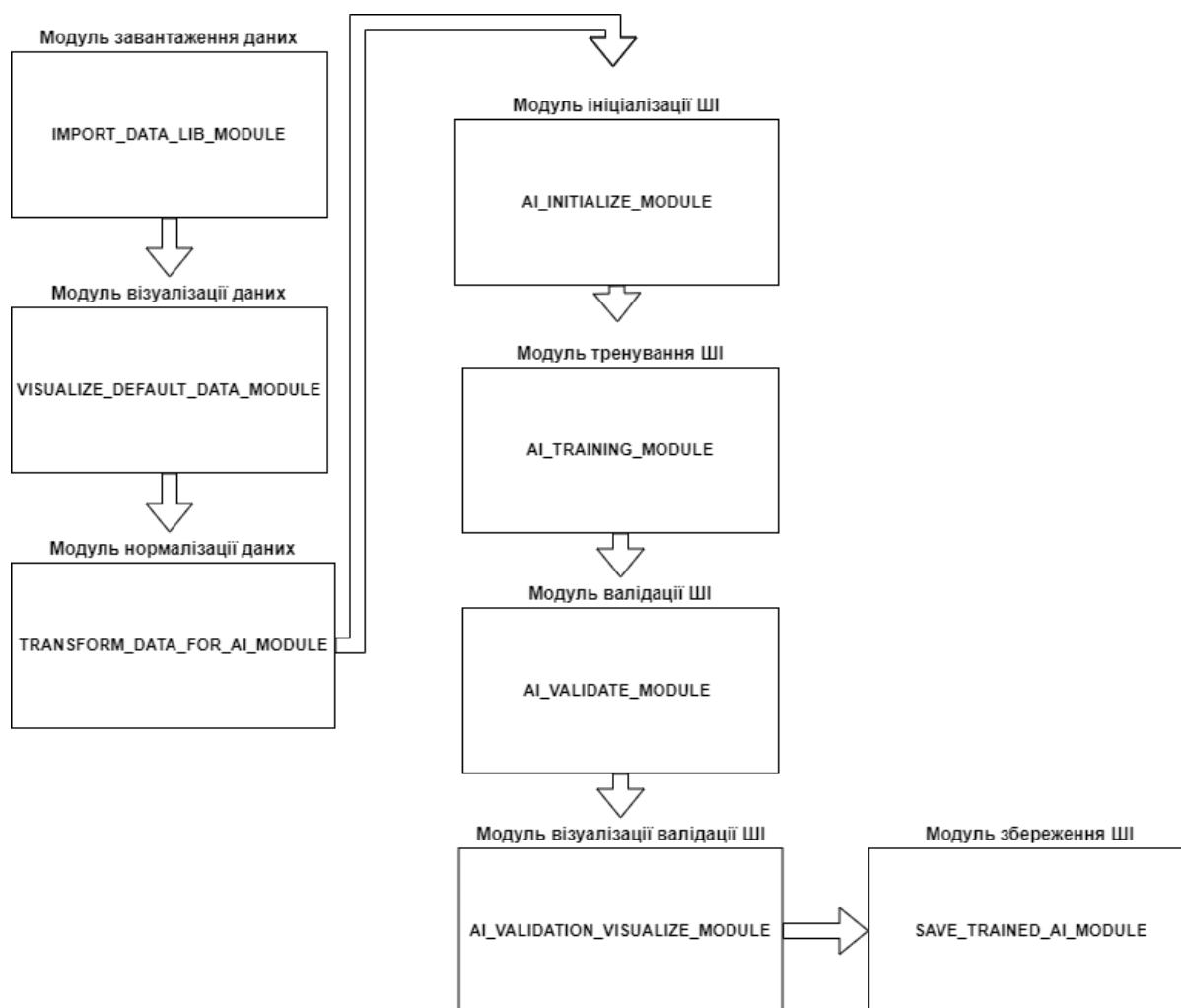


Рисунок 3.3 – Послідовність взаємодії програмних модулів у середовищі розробки моделі ШІ

У середовищі використання готової моделі штучного інтелекту послідовність взаємодії розроблених програмних модулів зображає рисунок 3.4.



Рисунок 3.4 – Взаємодія програмних модулів у середовищі використання готової моделі ШІ

3.2 Особливості реалізації способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

Для обробки даних, що застосовуються для тренування та валідації ШІ моделі використовується «csv» формат їх представлення (рис. 3.5).

```

1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""",4900.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00

```

Рисунок 3.5 – Приклад «csv» файлу

Завантажений набір даних містить такі початкові значення:

- «id» – унікальний ідентифікатор об'єкту
- «date» – дата продажу житла
- «price» – ціна житла
- «bedrooms» – кількість кімнат
- «bathrooms» – кількість ванних кімнат
- «sqft_living» – площа проживання
- «sqft_lot» – загальна площа, включаючи двір тощо
- «floors» – кількість поверхів
- «waterfront» – дані про те, чи розташоване житло на набережній
- «view» – інформація про огляд житла
- «condition» – загальний стан житла
- «grade» – оцінка житла
- «sqft_above» – площа вище 0 поверху
- «sqft_basement» – площа підвалу
- «yr_built» – рік, коли житло було побудоване
- «yr_renovated» – рік, коли житло було відремонтовано
- «zipcode» – поштовий індекс житла
- «lat» – широта, на якій розташоване житло
- «long» – довгота, на якій розташоване житло
- «sqft_living15» – площа проживання, станом на 2015 рік
- «sqft_lot15» – загальна площа, станом на 2015 рік

Для зручності використання та простої взаємодії модулів між собою використовується формат «Jupyter Notebook» та розширення файлу «.ipynb». Це дозволяє виконувати програмний код у своєрідних блоках, що робить зручним відображення графіків та взаємодію із моделлю в цілому.

В подальшому використанні готової моделі штучного інтелекту використовується стандартне розширення «.ру», що реалізує виконання коду у традиційному для мови програмування «Python» стилі.

Для збереження даних використовується локальне сховище, що представляє собою HDD або SSD диск.

Для збереження вимог стандартизації використовується файл із розширенням «bin», що зберігає дані у бінарному форматі.

При розробці системи використовувались мови програмування «Python» та «JavaScript», а саме певні бібліотеки, що допомагають у вирішенні задачі та відповідають поставленим вимогам та цілям і повністю їх задовольняють.

Основна частина системи використовує бібліотеки «Python», а саме:

- pandas
- matplotlib
- sklearn
- seaborn
- tensorflow
- numpy
- joblib

Для зрозумілості використання кожної з бібліотек та повному розумінні їх використання у проекті скористаємося схемами, що відображають бібліотеки та методи, що ці бібліотеки використовують. У схемах використаємо позначення типу «function()» – для функцій і методів бібліотеки, «field» – для об'єктів, класів, властивостей, полів бібліотеки.

На рисунку 3.6 зображено схему використання бібліотеки «pandas» та її бібліотек.

pandas
+read_csv()
+to_datetime()
+DataFrame()
+DataFrame.head()
+DataFrame.T
+DataFrame.info()
+DataFrame.describe()
+DataFrame.transpose()
+DataFrame.apply()
+DataFrame.groupby()
+DataFrame.mean()
+DataFrame.plot()
+DataFrame.isNull()
+DataFrame.sum()
+DataFrame.drop()
+DataFrame.values

Рисунок 3.6 – Використання бібліотеки «pandas» у розробленій системі

На рисунку 3.7 зображено схему використання бібліотеки «matplotlib» у системі.

matplotlib
pyplot
+figure()
+add_subplot()
+tight_layout()
+scatter()
+plot()

Рисунок 3.7 – Використання бібліотеки «matplotlib» у розробленій системі

На рисунку 3.8 зображено схему використання бібліотеки «sklearn» у розробленій системі.

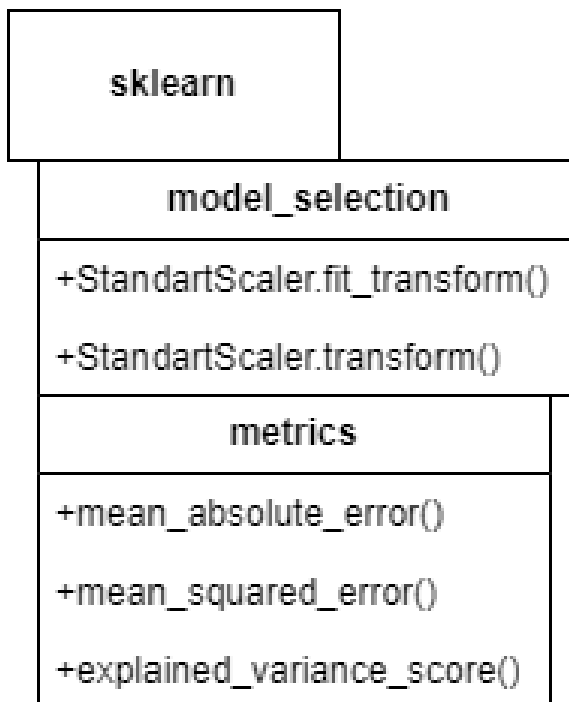


Рисунок 3.8 – Використання бібліотеки «sklearn» у розробленій системі

На рисунку 3.9 зображено схему використання бібліотеки та її методів «seaborn» при розробці програмної складової.

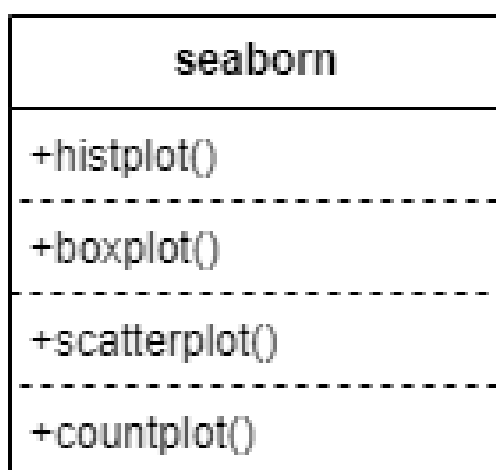


Рисунок 3.9 – Використання бібліотеки «seaborn» у розробленій системі

На рисунку 3.10 зображено схему використання бібліотеки «tensorflow».

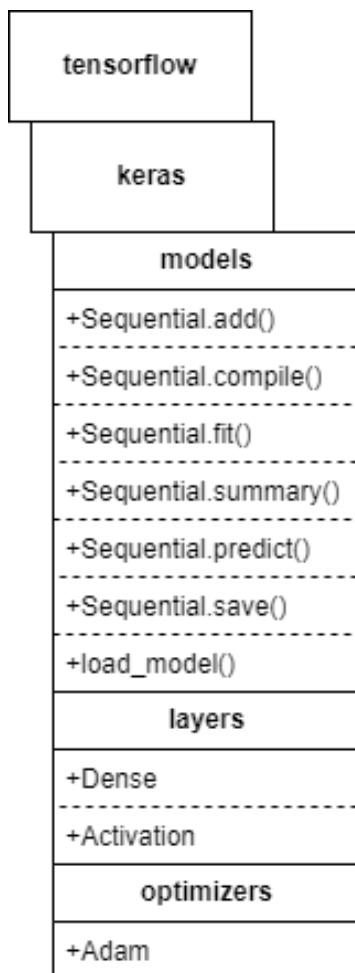


Рисунок 3.10 – Використання бібліотеки «tensorflow» у розробленій системі

На рисунку 3.11 зображено схему використання бібліотеки «numpy».

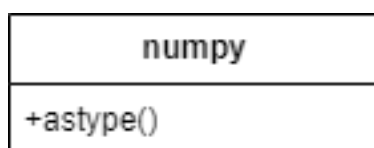


Рисунок 3.11 – Використання бібліотеки «numpy» у розробленій системі

На рисунку 3.12 зображено схему використання бібліотеки «joblib».

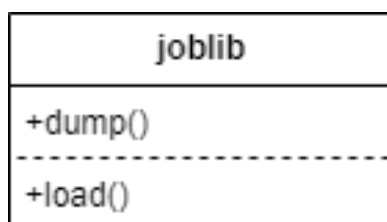


Рисунок 3.12 – Використання бібліотеки «joblib» у розробленій системі

Також використовується бібліотека «child_process» із середовища «Node» (рис. 3.13).

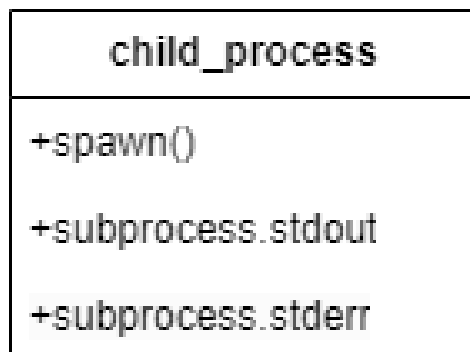


Рисунок 3.13 – Використання методів бібліотеки «child_process» у розробленій системі

Загальну схему використання усіх бібліотек, що наявні у розробленому додатку зображено на рисунку 3.14.

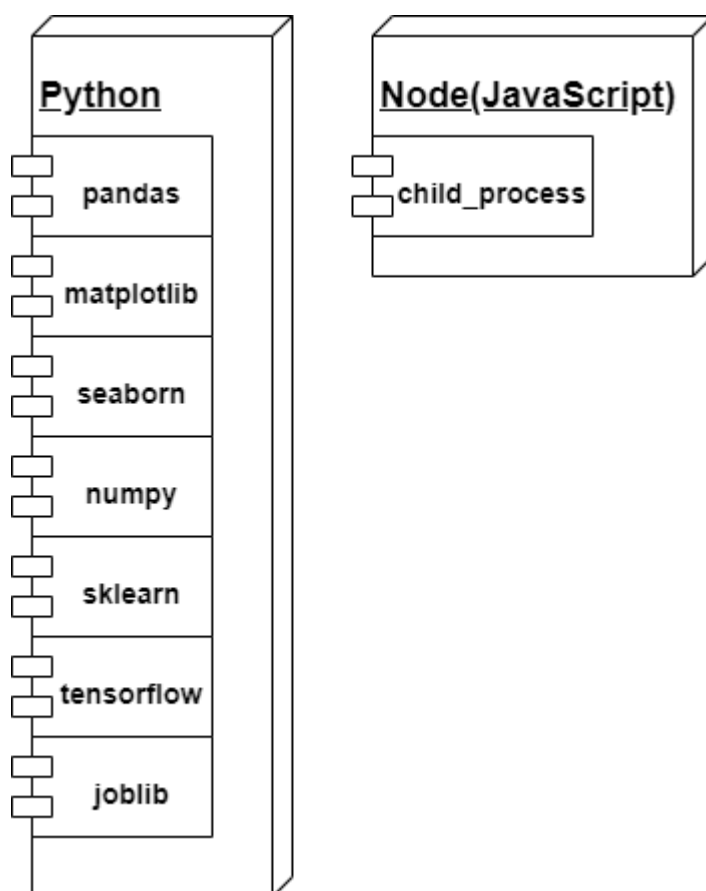


Рисунок 3.14 – Загальна схема бібліотек, які використовуються у розробленій системі прогнозування цін

3.3 Розробка програмних модулів

У розділі "Розробка програмних модулів" буде розглянуто детальніші аспекти розробки кожного модуля системи передбачення ціни житла методами штучного інтелекту.

Розробка програмних модулів є важливим етапом у створенні потужної системи передбачення цін житла, яка допомагатиме користувачам отримати цінну інформацію та зробити обґрунтовані рішення в галузі нерухомості.

При розробці програмних модулів системи було використано мову програмування «Python».

Модуль «IMPORT_DATA_LIB_MODULE» передбачений для завантаження даних. Програмна реалізація даного модуля передбачає імпорт даних із файлу, що містить дані про характеристики житла. Щоб коректно завантажити дані використовується бібліотека «pandas» та її метод «read_csv», що дозволяє завантажити дані з «.csv» файлу та привести до типу «DataFrame».

Результатом виконання даного модуля є виведення перших п'ятих рядків із характеристиками житла для кращого розуміння структури завантажених даних(рис. 3.15).

	0	1	2	3	4
id	7129300520	6414100192	5631500400	2487200875	1954400510
date	20141013T000000	20141209T000000	20150225T000000	20141209T000000	20150218T000000
price	221900.0	538000.0	180000.0	604000.0	510000.0
bedrooms	3	3	2	4	3
bathrooms	1.0	2.25	1.0	3.0	2.0
sqft_living	1180	2570	770	1960	1680
sqft_lot	5650	7242	10000	5000	8080
floors	1.0	2.0	1.0	1.0	1.0
waterfront	0	0	0	0	0
view	0	0	0	0	0
condition	3	3	3	5	3
grade	7	7	6	7	8
sqft_above	1180	2170	770	1050	1680
sqft_basement	0	400	0	910	0
yr_built	1955	1951	1933	1965	1987
yr_renovated	0	1991	0	0	0
zipcode	98178	98125	98028	98136	98074
lat	47.5112	47.721	47.7379	47.5208	47.6168
long	-122.257	-122.319	-122.233	-122.393	-122.045
sqft_living15	1340	1690	2720	1360	1800
sqft_lot15	5650	7639	8062	5000	7503

Рисунок 3.15 – Початковий вигляд даних

Наступний крок візуалізація даних та їх аналіз для подальшого використання у системі за допомогою модуля «VISUALIZE_DEFAULT_DATA_MODULE». Даний процес розбито на підмодулі, кожен з яких візуалізує та приводить у вигляд графіку або діаграми попередньо завантажені дані. Підмодуль «DIAGRAM_1_SUBMODULE» відображає типи даних у вигляді таблиці та транспонує їх, для цього використовуються методи бібліотеки «pandas»: «describe» – загальна інформація про раніше завантажений «DataFrame», «transpose» – для транспонування та відображення транспонованих даних.

Результатом роботи підмодуля «DIAGRAM_1_SUBMODULE» буде таблиця із назвами характеристик та їхнім типом відповідно, а також кількість «Non-Null» значень(рис. 3.16) та транспонування даних та виведення результату для подальшого аналізу(рис 3.17).

#	Column	Non-Null	Count	Dtype
0	id	21613	non-null	int64
1	date	21613	non-null	object
2	price	21613	non-null	float64
3	bedrooms	21613	non-null	int64
4	bathrooms	21613	non-null	float64
5	sqft_living	21613	non-null	int64
6	sqft_lot	21613	non-null	int64
7	floors	21613	non-null	float64
8	waterfront	21613	non-null	int64
9	view	21613	non-null	int64
10	condition	21613	non-null	int64
11	grade	21613	non-null	int64
12	sqft_above	21613	non-null	int64
13	sqft_basement	21613	non-null	int64
14	yr_built	21613	non-null	int64
15	yr_renovated	21613	non-null	int64
16	zipcode	21613	non-null	int64
17	lat	21613	non-null	float64
18	long	21613	non-null	float64
19	sqft_living15	21613	non-null	int64
20	sqft_lot15	21613	non-null	int64

Рисунок 3.16 – Таблиця відображення типів даних

	count	mean	std	min	25%	50%	75%	max
id	21613.0	4.580302e+09	2.876566e+09	1.000102e+06	2.123049e+09	3.904930e+09	7.308900e+09	9.900000e+09
price	21613.0	5.400881e+05	3.671272e+05	7.500000e+04	3.219500e+05	4.500000e+05	6.450000e+05	7.700000e+06
bedrooms	21613.0	3.370842e+00	9.300618e-01	0.000000e+00	3.000000e+00	3.000000e+00	4.000000e+00	3.300000e+01
bathrooms	21613.0	2.114757e+00	7.701632e-01	0.000000e+00	1.750000e+00	2.250000e+00	2.500000e+00	8.000000e+00
sqft_living	21613.0	2.079900e+03	9.184409e+02	2.900000e+02	1.427000e+03	1.910000e+03	2.550000e+03	1.354000e+04
sqft_lot	21613.0	1.510697e+04	4.142051e+04	5.200000e+02	5.040000e+03	7.618000e+03	1.068800e+04	1.651359e+06
floors	21613.0	1.494309e+00	5.399889e-01	1.000000e+00	1.000000e+00	1.500000e+00	2.000000e+00	3.500000e+00
waterfront	21613.0	7.541757e-03	8.651720e-02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
view	21613.0	2.343034e-01	7.663176e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	4.000000e+00
condition	21613.0	3.409430e+00	6.507430e-01	1.000000e+00	3.000000e+00	3.000000e+00	4.000000e+00	5.000000e+00
grade	21613.0	7.656873e+00	1.175459e+00	1.000000e+00	7.000000e+00	7.000000e+00	8.000000e+00	1.300000e+01
sqft_above	21613.0	1.788391e+03	8.280910e+02	2.900000e+02	1.190000e+03	1.560000e+03	2.210000e+03	9.410000e+03
sqft_basement	21613.0	2.915090e+02	4.425750e+02	0.000000e+00	0.000000e+00	0.000000e+00	5.600000e+02	4.820000e+03
yr_built	21613.0	1.971005e+03	2.937341e+01	1.900000e+03	1.951000e+03	1.975000e+03	1.997000e+03	2.015000e+03
yr_renovated	21613.0	8.440226e+01	4.016792e+02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.015000e+03
zipcode	21613.0	9.807794e+04	5.350503e+01	9.800100e+04	9.803300e+04	9.806500e+04	9.811800e+04	9.819900e+04
lat	21613.0	4.756005e+01	1.385637e-01	4.715590e+01	4.747100e+01	4.757180e+01	4.767800e+01	4.777760e+01
long	21613.0	-1.222139e+02	1.408283e-01	-1.225190e+02	-1.223280e+02	-1.222300e+02	-1.221250e+02	-1.213150e+02
sqft_living15	21613.0	1.986552e+03	6.853913e+02	3.990000e+02	1.490000e+03	1.840000e+03	2.360000e+03	6.210000e+03
sqft_lot15	21613.0	1.276846e+04	2.730418e+04	6.510000e+02	5.100000e+03	7.620000e+03	1.008300e+04	8.712000e+05

Рисунок 3.17 – Транспоновані дані

Підмодуль «DIAGRAM_2_SUBMODULE» зображує за допомогою діаграми відношення кількості будинків до ціни (рис. 3.18).

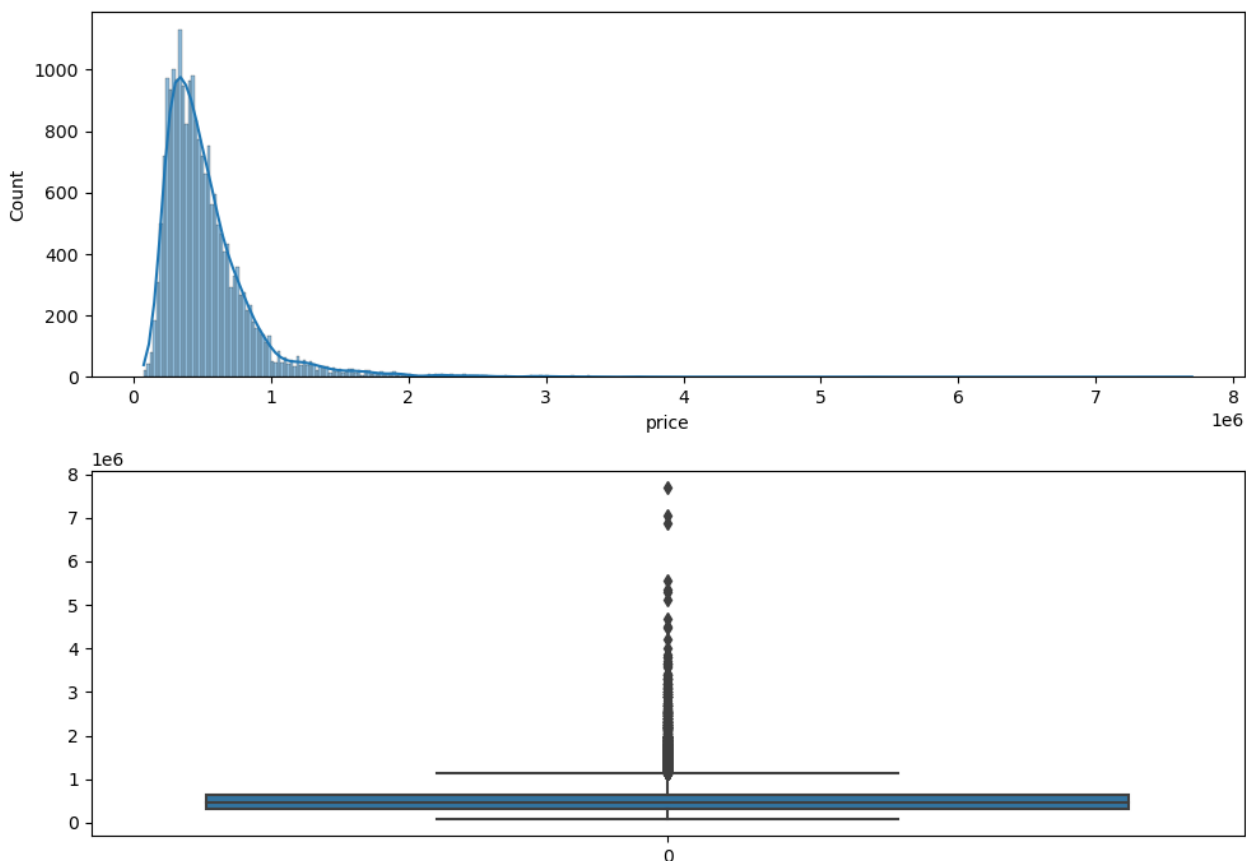


Рисунок 3.18 – діаграма відношення кількості будинків до ціни

Для цього використовується бібліотека «matplotlib.pyplot» та її методи: «figure» – для створення своєрідної площини, де будуть відображатися внесені в подальшому дані, «add_subplot» – додає вісь до заданої площини, «tight_layout» – регулює відступи між графіками. А також бібліотека «seaborn» та її методи: «histplot» – будує гістограму для заданих даних, «boxplot» – будує блочну діаграму для заданих даних.

Підмодуль «DIAGRAM_3_SUBMODULE» відображає відношення ціни до різних характеристик житла, що відображають інформацію про площу житла. А саме ціна до загальної площі, площі проживання, наземної площі та підземної(підвалу). Для цього використовується метод бібліотеки «seaborn» – «scatterplot», що відображає точкову діаграму, із семантичним групуванням.

Рисунок 3.19 відображає візуалізовані діаграми.

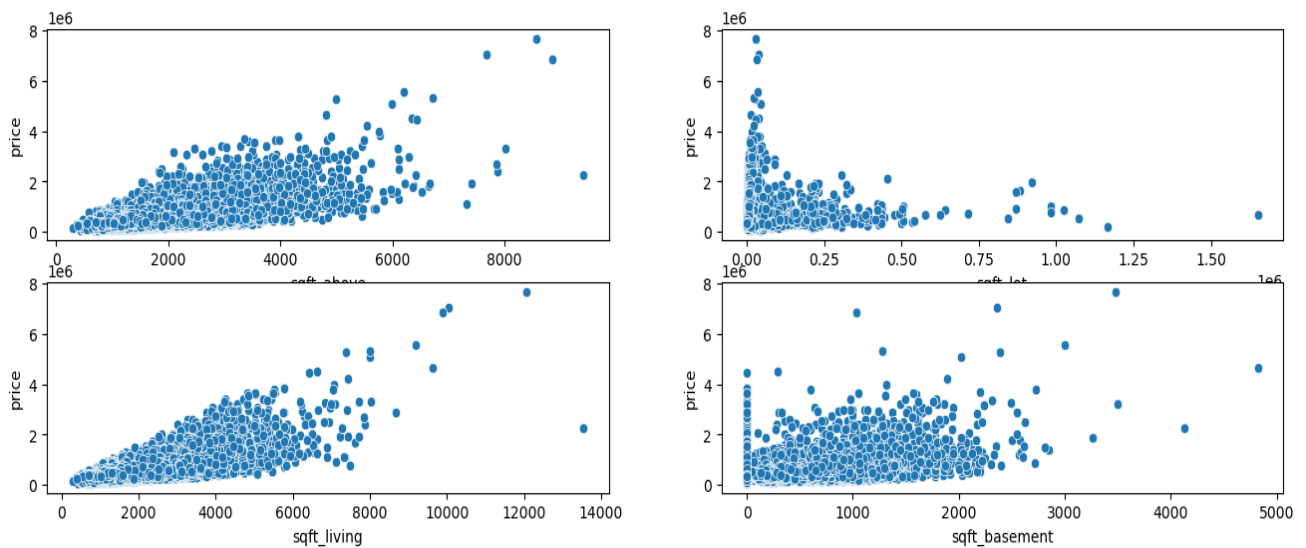


Рисунок 3.19 – Діаграми відношення кількості ціни до площі

Підмодуль «DIAGRAM_4_SUBMODULE» відображає статистику наявних даних, таких як кількість будинків із характеристиками: кількість кімнат, кількість поверхів, кількість ванних кімнат, оцінка житла. Для цього використовується метод бібліотеки «seaborn» – «countplot», що відображає кількість спостережень(даних) у заданих інтервалах за допомогою стовпчикової діаграми.

Рисунок 3.20 демонструє результат виконання Підмодуля «DIAGRAM_4_SUBMODULE».

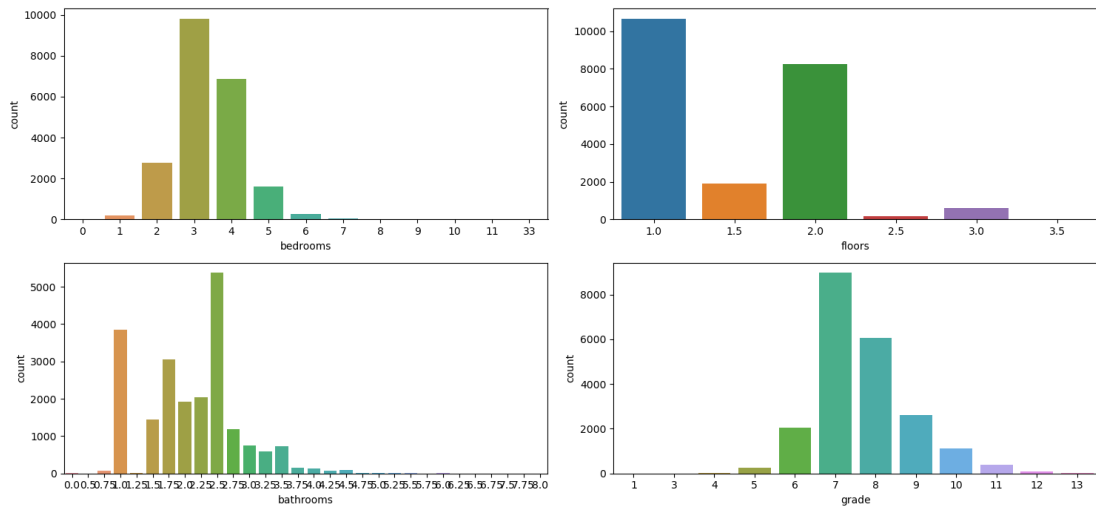


Рисунок 3.20 – Стовпчикові діаграми кількості характеристик до наявних даних

Підмодуль «DIAGRAM_5_SUBMODULE» реалізує одразу декілька функцій. Перша з яких – приведення даних до вигляду, необхідному для їх відображення, друга функція – безпосередньо відображення необхідної діаграми.

Результатом відпрацювання цієї функції є створення колонок «month» та «year», взятих на основі існуючої колонки «date» для аналізу даних пов'язаних з нею.

Реалізація другої функції використовує методи бібліотеки «pandas»: «groupby» – для групування заданих значень, «plot» – для виведення графіку на основі заданих даних, «mean» – повертає середнє значення для заданої осі або колонки.

Результатом відпрацювання підмодуля «DIAGRAM_5_SUBMODULE» є графіки відношення ціни до місяця та до року відповідно(рис. 3.21).

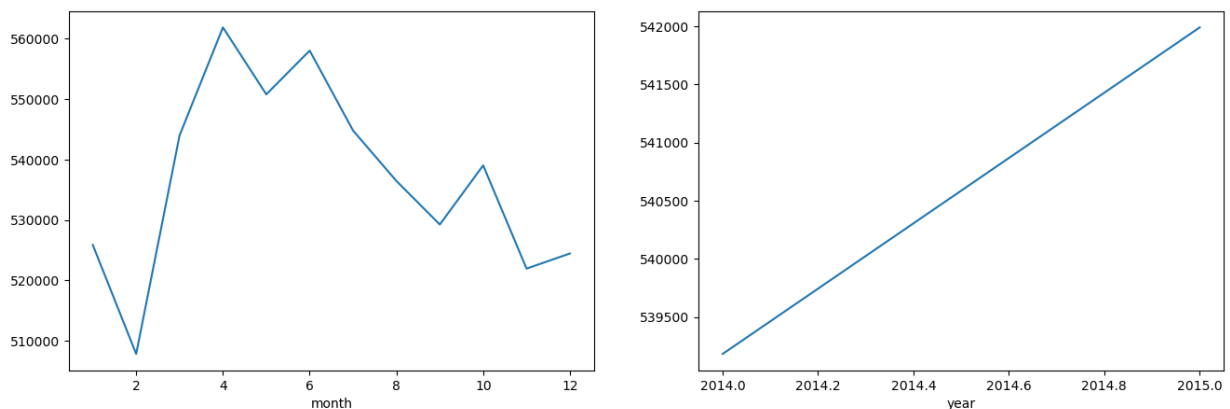


Рисунок 3.21 – Графіки відношення ціни до місяця та до року

Наступний модуль «TRANSFORM_DATA_FOR_AI_MODULE» відповідає за нормалізацію даних та їх підготовку до застосування. Для цього використовуються бібліотеки «pandas», «numpy», «sklearn».

Методи бібліотеки «pandas»:

- isNull – перевіряє дані на вміст «Null» значень
- sum – виводить суму значень по заданій осі, колонці
- drop – видаляє задану вісь, колонку

Та властивість об'єкту «DataFrame» – «values», що переводить дані до типу «ndarray» необхідного при роботі із ШІ моделлю.

Методи бібліотеки «numpy» – «astype», що створює копію масиву, приведеного до заданого типу.

Методи бібліотеки «sklearn»:

- model_selection.train_test_split – розділяє дані на тренувальну та тестову вибірки за заданими параметрами
- preprocessing.fit_transform – стандартизує дані згідно формули (3.1) та задає вимоги до майбутніх стандартизацій:
- preprocessing.transform – стандартизує дані згідно попередньо заданих для цього вимог.

$$z = \frac{(x - u)}{s}. \quad (3.1)$$

Результатом виконання є стандартизовані дані, із якими повинна працювати модель штучного інтелекту, та які повністю відповідають цим вимогам.

Модуль ініціалізації ШІ моделі «AI_INITIALIZE_MODULE» використовує бібліотеку «tensorflow», а саме «keras.models.Sequential» для ініціалізації моделі та метод «add» в поєднанні з параметром «Dense» та «Activation» із «keras.layers» – для додавання шарів то моделі штучного інтелекту, що є багатошаровим перцептроном. Метод «compile» використовується для налаштування моделі для тестування і містить у собі параметри «optimizer», що

визначає оптимізатор для моделі(у нашому випадку це вбудований у бібліотеку, що ми використовуємо оптимізатор «Adam» із «keras.optimizers») та «loss», що визначає, як буде вимірюватись похибка. Для обрахування похибки обрано параметр «mse», що використовує середньоквадратичну похибку для обрахувань.

У результаті виконання даного модуля, ініціалізується модель ШІ та задаються усі необхідні параметри необхідні для початку роботи з нею.

Модуль тренування моделі штучного інтелекту «AI_TRAINING_MODULE» використовує метод «fit» та «summary» із бібліотеки «tensorflow.keras». Метод «fit» виконує навчання раніше ініціалізованої моделі штучного інтелекту за заданими параметрами. В якості параметрів він використовує:

- «epochs», що задають кількість ітерацій проведених над набором даних.
- «x» – параметр, що задає вхідні дані для тренування
- «y» – параметр, що задає цільові параметри для передбачення
- «validation_data» – дані по яких буде розраховуватись раніше задана похибка
- «batch_size» – розмір вибірки для кожної епохи(ітерації)

Метод «summary» виводить результат відпрацювання тестування моделі штучного інтелекту.

Результат виконання даного модулю зображено на рисунку 3.22.

```

Epoch 1/400
114/114 [=====] - 2s 7ms/step - loss: 425258680320.0000 - val_loss: 428754731008.0000
Epoch 2/400
114/114 [=====] - 0s 4ms/step - loss: 423355154432.0000 - val_loss: 421869584384.0000
Epoch 3/400
114/114 [=====] - 0s 4ms/step - loss: 398061469696.0000 - val_loss: 362838622208.0000
Epoch 4/400
114/114 [=====] - 0s 4ms/step - loss: 285520461824.0000 - val_loss: 191845040128.0000
Epoch 5/400
114/114 [=====] - 0s 4ms/step - loss: 132763918336.0000 - val_loss: 91028496384.0000
Epoch 6/400
114/114 [=====] - 1s 5ms/step - loss: 85843804160.0000 - val_loss: 72622530560.0000
Epoch 7/400
114/114 [=====] - 0s 4ms/step - loss: 72315428864.0000 - val_loss: 63583653888.0000
Epoch 8/400
114/114 [=====] - 1s 5ms/step - loss: 65185325056.0000 - val_loss: 58774650880.0000
Epoch 9/400
114/114 [=====] - 1s 6ms/step - loss: 60818370560.0000 - val_loss: 55222038528.0000
Epoch 10/400
114/114 [=====] - 1s 5ms/step - loss: 57490173952.0000 - val_loss: 52413747200.0000
Epoch 11/400
114/114 [=====] - 1s 5ms/step - loss: 54633287680.0000 - val_loss: 50030202880.0000
Epoch 12/400
114/114 [=====] - 1s 5ms/step - loss: 52225896448.0000 - val_loss: 47923306496.0000
Epoch 13/400
...
Total params: 1,540
Trainable params: 1,540
Non-trainable params: 0

```

Рисунок 3.22 – Підсумок тестування моделі ШІ

Щоб визначити якість відпрацювання даного модуля, використовується додатковий підмодуль «AI_TRAINING_LOSS_SUBMODULE», що відображає у вигляді графіка кривої, відношення відсотку похибки до кількості ітерацій. Для цього дані «model.history.history», що відображають дані відпрацювання попереднього модуля тестування, приводяться до раніше описаного типу «DataFrame», та на основі цих даних будується графік.

Модулі валідації «AI_VALIDATE_MODULE» та візуалізації валідації «AI_VALIDATION_VISUALIZE_MODULE» об'єднані у блок для зручності реалізації. Цей блок містить у собі бібліотеку «sklearn.metrics» та використовує такі її методи:

- «mean_absolute_error» – відображає середню абсолютну похибку
- «mean_squared_error» – відображає середньоквадратичну похибку
- «explained_variance_score» – відображає середній відсоток правильності

прогнозувань

Та бібліотеку «numpy» і її метод «sqrt», що повертає невід'ємний квадратний корінь масиву, згідно елементів у ньому.

А також бібліотеку «matplotlib.pyplot» та її методи «scatter» та «plot» для об'єднаного відображення:

- Точкової діаграми передбачень
- Лінійної прямої ідеальних передбачень

Результатом виконання є відображення таких значень:

- MAE – середня абсолютна похибка
- MSE – середньоквадратична похибка
- RMSE – кореневе середньоквадратичне відхилення
- VarScore – відсоток точності передбачення

А також співставленні діаграми, що візуально демонструють точність передбачення та різницю прогнозованих значень цін до фактичних значень із валідаційної вибірки, тобто тих, де похибка дорівнює нулю.

Модуль «SAVE_TRAINED_AI_MODULE» зберігає натреновану модель штучного інтелекту із всіма значеннями, для подальшого використання. Цей

модуль використовує метод бібліотеки «tensorflow» – «save», що зберігає модель у вигляді файлу «Tensorflow SavedModel»(рис. 3.23). Та бібліотеку «joblib» та її метод «dump», що зберігає будь який об'єкт Python в один «.bin» файл. У нашому випадку «dump» використовується для збереження вимог стандартизації даних, для подальшого використання цих вимог для трансформації незалежних від середовища даних.

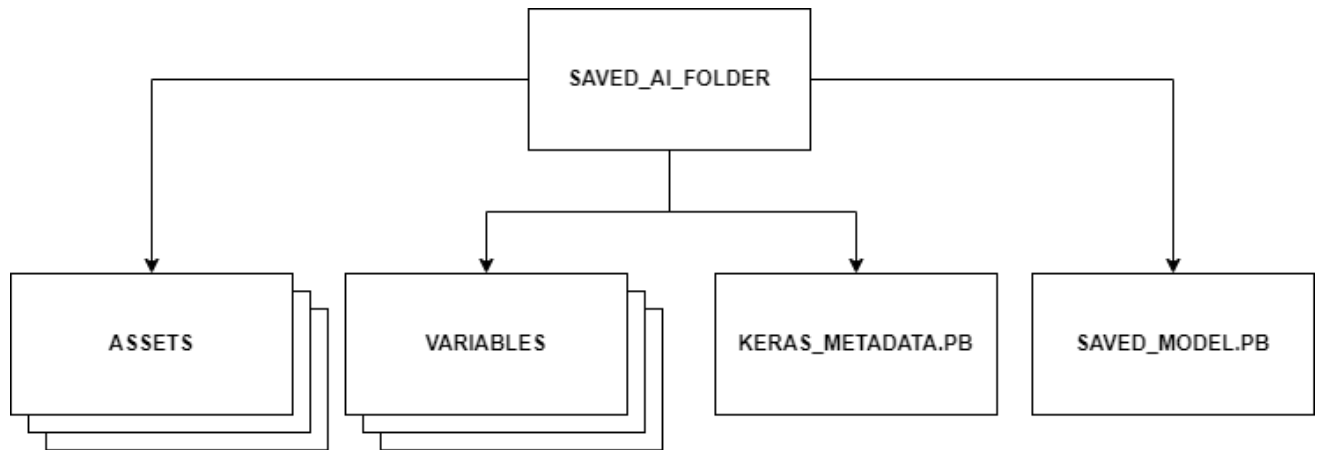


Рисунок 3.23 – Структура збереженої моделі штучного інтелекту

Структура містить:

Папку «assets» – зовнішні дані, що потрібні для процесу передбачення

Папку «variables» – містить змінні, що використовуються у моделі ШІ

Файл «keras_metadata.pb» – містить метадані про модель штучного інтелекту

Файл «saved_model.pb» – безпосередньо збережена модель

Модель такої структури із раніше оголошеними даними і є результатом виконання модуля «SAVE_TRAINED_AI_MODULE».

Модуль «LOAD_TRAINED_AI_MODULE» є модулем, що передбачає завантаження збереженої моделі, для подальшого її використання у незалежному від попереднього середовищі. Для цього використовується метод бібліотеки «tensorflow.keras» – «load_model», для завантаження наявної збереженої моделі штучного інтелекту за вказаним шляхом. А також раніше згадана бібліотека «joblib» та метод «load», що відтворює Python об'єкт створений за допомогою іншого методу «dumb». Результатом відпрацювання даного модуля буде

завантажена та готова до використання, попередньо натренована модель штучного інтелекту та імпортовані вимоги щодо стандартизації даних.

Модуль «AI_NODE_INTEGRATION_MODULE» відповідає за інтеграцію та успішну взаємодію між виконуваним Python файлом та середовищем Node, необхідним для створення локального сервера, з яким в подальшому взаємодіє клієнт за допомогою HTTP запитів. Цей модуль для чіткості розуміння поділено на дві частини: частина написана на мові Python для взаємодії із Node (Частина №1), частина написана на мові JavaScript, а саме на платформі Node для взаємодії із Python (Частина №2).

Частина №1 використовує вбудовану бібліотеку «sys» та значення «argv» для отримання списку аргументів, що були передані через «command line» та метод «print» із параметром та його значенням – «flush=True», що відключає буферизацію та змушує скрипт дочекатись завершення і тільки потім вивести результат. Після отримання аргументів, вони стандартизуються відповідно до раніше завантажених вимог за допомогою методу «transform» об'єкту «StandartScaler» із бібліотеки «sclearn.preprocessing». Стандартизовані дані передаються у модель штучного інтелекту та робиться передбачення за допомогою методу «predict» із бібліотеки «tensorflow». Результатом роботи частина №1 буде повернення прогнозованої ціни.

Частина №2 використовує метод «spawnSync» із вбудованої у Node бібліотеки «child_process», що створює новий процес у середовищі та дозволяє із ним взаємодіяти. «spawnSync» приймає у себе такі параметри:

- «command» – опис команди, що потрібно запустити, у нашому випадку це команда «python», що буде виконувати подальші дії у розумінні однойменної мови програмування.

- «args» – перелік аргументів, першим аргументом ми передаємо шлях до потрібно нам файлу, тобто Частини №1, решта аргументів – це характеристики житла.

Після відпрацювання створеного процесу, ми отримуємо дані про його відпрацювання: «subprocess.stdout» повертає результат у разі успішного відпрацювання процесу, «subprocess.stderr» повертає опис помилки у разі

провалення відпрацювання процесу. Отримані значення приводяться до зрозумілого для людини вигляду за допомогою статичного методу JavaScript «toString», що приводить дані у строковий тип «string». Результатом відпрацювання Частини №2 є повернення результату, готового до відправки клієнту.

Схематично принцип взаємодії частин у модулі зображено на рисунку 3.24

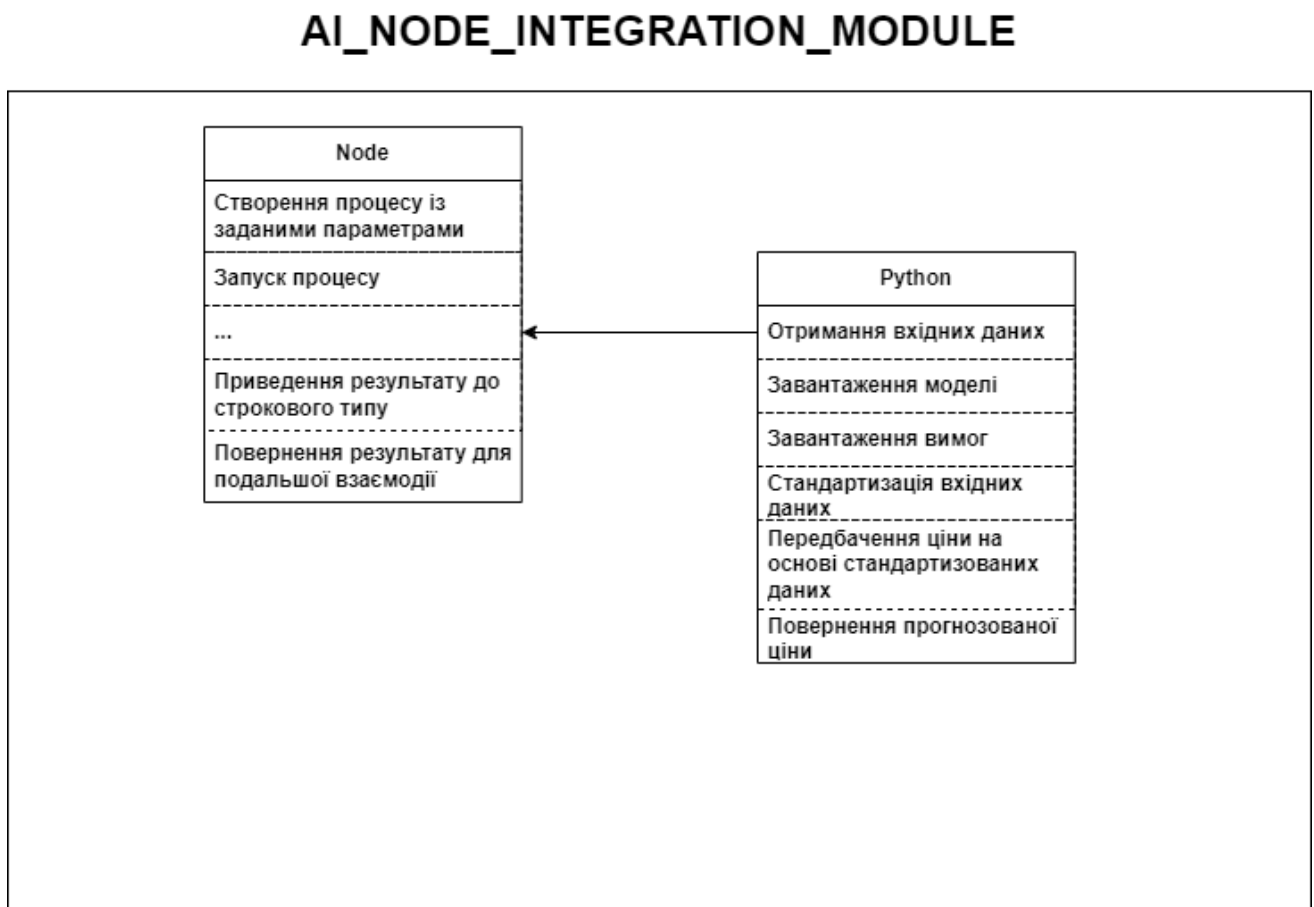


Рисунок 3.24 – Схема принципу взаємодії виокремлених частин у модулі

Модуль «NEST_STANDALONE_SERVER_MODULE» виконує функцію створення локального сервера для подальшої реалізації зручної взаємодії клієнта із ШІ моделлю. Даний модуль представляє собою програмний додаток, побудований за допомогою фреймворку «Nest». За допомогою нього створено контроллер, що взаємодіє із клієнтом за допомогою створених «REST endpoints», до яких клієнт може робити HTTP запит та отримувати заданий у контролері результат. Створений контроллер взаємодіє із попереднім модулем

«AI_NODE_INTEGRATION_MODULE» та передає результат на клієнтську частину у випадку звернення до нього клієнтом. Структуру даного модуля зображено на рисунку 3.25

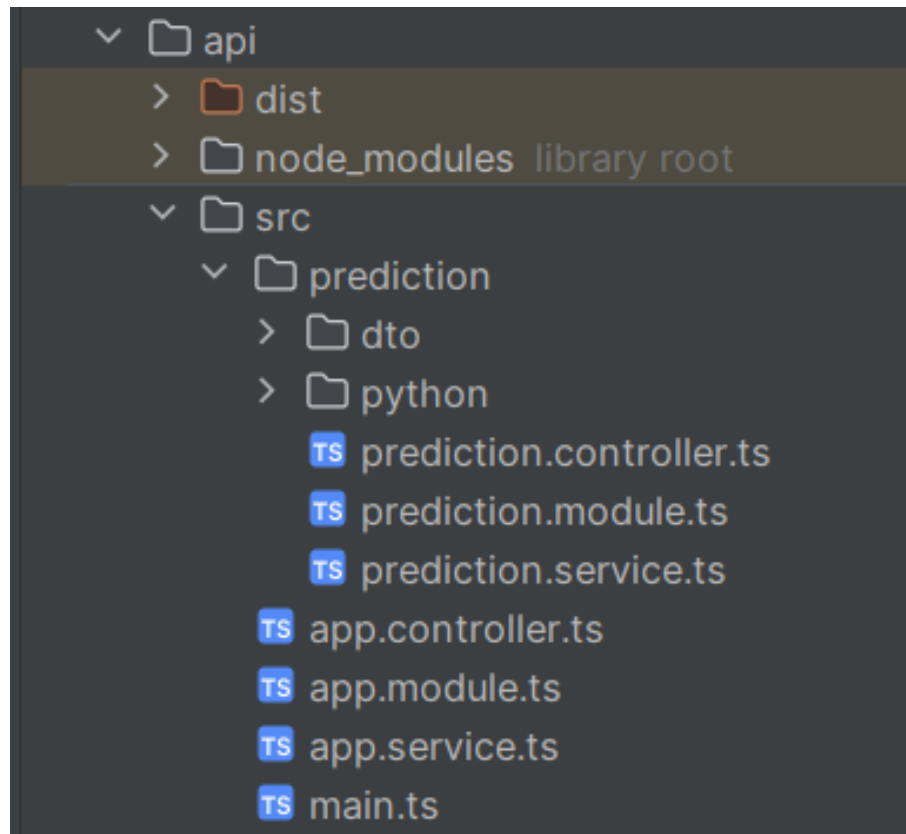


Рисунок 3.25 – Структура модуля «NEST_STANDALONE_SERVER_MODULE»

Останній із модулів «UI_MODULE» є реалізацією клієнтської частини розробленої системи. Даний модуль розроблений за допомогою бібліотеки «React.js» та допоміжних бібліотек:

- «axios» – для створення запитів до серверної частини (модуля «NEST_STANDALONE_SERVER_MODULE»)

- «@reduxjs/toolkit» – для організації контролю даних на клієнтській стороні системи

- «react-redux» – для взаємодії бібліотеки «@reduxjs/toolkit» з «React.js»

- «antd» – для розробки макету веб-застосунку та налаштування його зовнішнього вигляду

Результатом розробки модуля «UI_MODULE» є веб-застосунок, повністю функціональний та легкий для розуміння кінцевим користувачем (рис. 3.26).

The screenshot shows a web form with the following fields and values:

* Bedrooms	* View	* Yr_renovated
3	0	0
* Bathrooms	* Condition	* Lat
1.0	3	47.5112
* Sqft_living	* Grade	* Long
1180	7	-122.257
* Sqft_lot	* Sqft_above	* Sqft_living15
5650	1180	1340
* Floors	* Sqft_basement	* Sqft_lot15
1.0	0	5650
* Waterfront	* Yr_built	
0	1955	

At the bottom of the form is a blue button labeled "Predict price".

Рисунок 3.26 – Веб-застосунок, модуль «UI_MODULE»

Відображення процесу передбачення є важливою частиною для розуміння кінцевим користувачем, його очікування на результат та виконано у вигляді зображення процесу завантаження (рис. 3.27).

This screenshot shows the same form as in Figure 3.26, but with a loading indicator below the "Predict price" button. The indicator consists of three blue dots and the text "Making prediction..."

Рисунок 3.27 – Відображення процесу передбачення

3.4 Валідація способу предбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

Графік, зображений на рисунку 3.28, демонструє відношення значення функції втрат на тренувальній вибірці – «loss» та на валідаційній – «val_loss» до кількості ітерацій –«epochs».

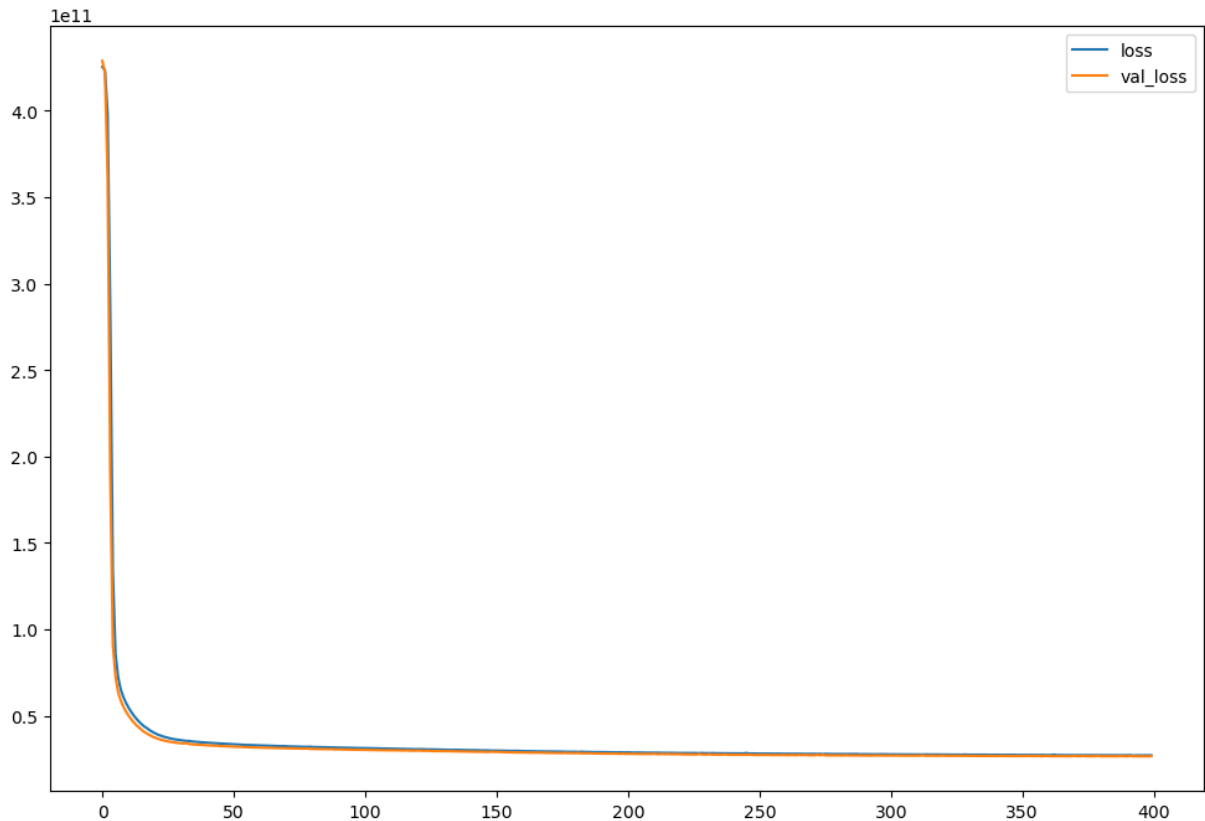


Рисунок 3.28 – Графік відношення відсотку похибки до кількості ітерацій

Для обрахування функції втрат застосовується середньоквадратична похибка. Із даного графіку видно, що із збільшенням ітерацій значення похибок значно зменшується, що свідчить про здатність моделі до тренування та засвоєння опрацьованих та проаналізованих нею даних.

Для визначення точності прогнозу та відсотку похибки, застосовується обрахування середньої абсолютної помилки, середньої квадратичної помилки, кореневого середньоквадратичного відхилення та відношення дисперсії валідаційної вибірки до тестової (рис. 3.29).

```

MAE: 102506.90180542899
MSE: 26652574652.81613
RMSE: 163256.16267944107
VarScore: 0.8063718200824314

```

Рисунок 3.29 – Показники точності передбачення

На рисунку 3.29 відображено показники точності передбачення:

– «MAE» – середня абсолютна помилка, що демонструє абсолютну різницю прогнозованих цін в середньому у ~ 102507 у.о. від фактичних.

– «MSE» – середня квадратична помилка, що відображає середній квадрат різниці прогнозованих цін до фактичних та дорівнює ~ 26652574653 .

– «RMSE» – кореневе середньоквадратичне відхилення, що відображає квадратний корінь середньоквадратичної помилки та дорівнює ~ 163256 у.о.

«VarScore» – відношення дисперсії валідаційної вибірки до тренувальної, демонструє точність прогнозу, де 1 – 100% точності. Значення дорівнює ~ 0.8 та означає, що точність моделі становить $\sim 80\%$.

Для зрозумілості та наглядного прикладу точності прогнозів створено графік (рис 3.30), що демонструє фактичні значення цін валідаційної вибірки та прогнозованні моделлю штучного інтелекту.

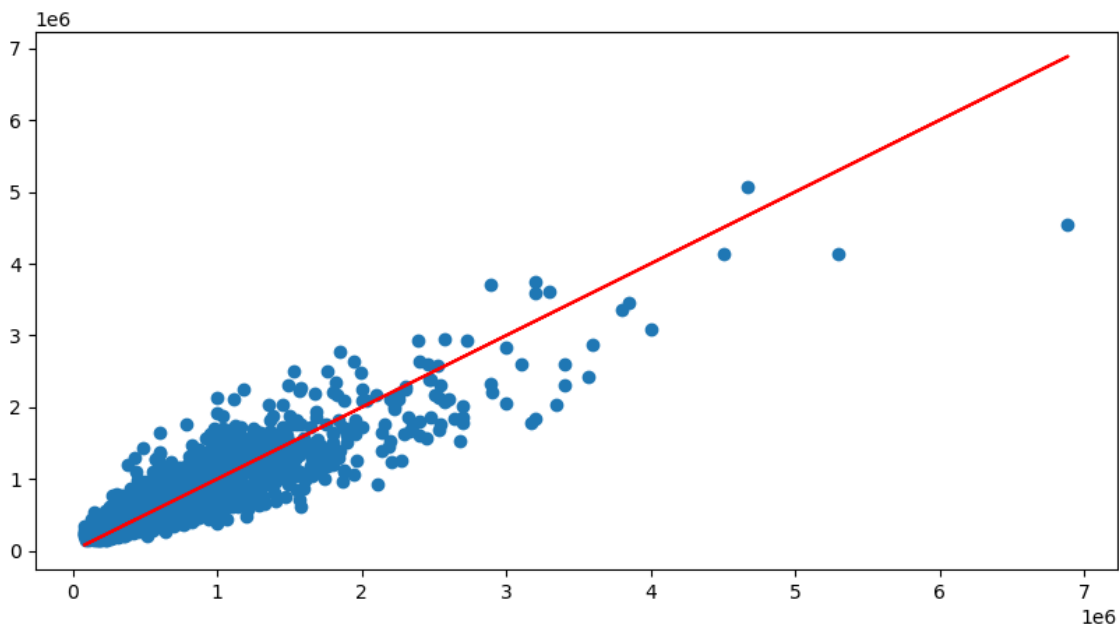


Рисунок 3.30 – Співставлені діаграми для оцінки точності передбачення

Отже, точність моделі ШІ є близько 80%, а обраховані похибки мають значення, що повністю задовольняє вимоги. Це свідчить про правильність процесу тренування моделі та її здатність до передбачення на нових, невідомих даних.

При використанні натренованої моделі, користувач, що взаємодіє з нею через клієнтський інтерфейс має змогу, при внесенні даних житла, отримати результат – ціну житла, точність якої буде не менше 80% (рис. 3.31).

* Bedrooms	* View	* Yr_renovated
3	0	0
* Bathrooms	* Condition	* Lat
1.0	3	47.5112
* Sqft_living	* Grade	* Long
1180	7	-122.257
* Sqft_lot	* Sqft_above	* Sqft_living15
5650	1180	1340
* Floors	* Sqft_basement	* Sqft_lot15
1.0	0	5650
* Waterfront	* Yr_built	
0	1955	

Predict price

274608.44 \$

Рисунок 3.31 – Відображення прогнозованої ціни

3.5 Висновки до розділу 3

Виконано програмну реалізацію способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах із визначенням структури модулів програмної системи та їх взаємозв'язок.

При розробці програмних модулів системи було використано мову програмування «Python». Програмна система складається з модулів, які

вирішують задачі ініціалізації необхідних даних, їх аналізу та валідації, передбачення засобами штучного інтелекту, а саме за допомогою створеного багат шарового перцептронну, а саме для передбачення ціни житла у розподілених ріелторських системах.

Виконана валідація моделі підтвердила її коректність, похибки перевірки на тестовій виборці склали: ~102507 у.о. – для середньої абсолютної помилки, ~26652574653 – для середньої квадратичної помилки, ~163256 у.о. – для кореневого середньоквадратичного відхилення, точність моделі становить ~80%.

Висновки

В кваліфікаційній роботі створено спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

Для досягнення поставленої мети було вирішено такі задачі дослідження: проаналізовано сучасні способи передбачення ціни житла у ріелторських системах; виконано аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем; реалізована інформаційна технологія способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах; проведено експериментальне тестування інформаційної технології.

При реалізації способу передбачення ціни житла засобами штучного інтелекту у розподільних ріелторських системах визначено загальну послідовність задачі передбачення, що включає структуру собівартості, з подальшою імплементацією методів штучного інтелекту урахуванням особливостей їх застосування.

Для вирішення задачі передбачення методами штучного інтелекту було використано багатосаровий перцептрон. Визначено фактори та чинники, що впливають на результат передбачення з використанням методів штучного інтелекту та визначено номенклатуру параметрів контролю показників передбачення, розроблено план дій до кожного із випадків.

Розроблена ріелторська система реалізована як розподілена система та включає функціональну структуру системи та структурний вигляд програмних модулів. Описано та поставлено межі відповідальності кожного із модулів, блоків та елементів системи. Виконано програмну реалізацію способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах із визначенням структури модулів програмної системи та їх взаємозв'язок.

Програмна система складається з модулів, які вирішують задачі ініціалізації необхідних даних, їх аналізу та валідації, передбачення засобами штучного інтелекту, а саме за допомогою створеного багатосарового

перцептрон, а саме для передбачення ціни житла у розподілених ріелторських системах.

Виконана валідація моделі підтвердила її коректність, похибки перевірки на тестовій виборці склали: ~102507 у.о. – для середньої абсолютної помилки, ~26652574653 – для середньої квадратичної помилки, ~163256 у.о. – для кореневого середньоквадратичного відхилення, точність моделі становить ~80%.

Перелік посилань

1. Aggarwal, C. C. Neural networks and deep learning [Text] – Springer, 2018 – V. 10 – P. 978-983.
2. A Simple Overview of Multilayer Perceptron (MLP) Deep Learning [Електронний ресурс]. – Режим доступу: <https://www.analyticsvidhya.com/blog/2020/12/mlp-multilayer-perceptron-simple-overview/>
3. Data processing using artificial neural networks [Електронний ресурс]. – Режим доступу: <https://www.intechopen.com/chapters/71673>
4. Dom.ria [Електронний ресурс] – Режим доступу до ресурсу: <https://dom.ria.com/uk/>
5. Ferlan, N.; Bastic, M.; Psunder, I. Influential Factors on the Market Value of Residential Properties. Inz. Ekon. Eng. Econ. 2017, 28, 135–144.
6. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow / Aurélien Géron., 2017. – (O'Reilly Media). – (1st Edition) –110 p.
7. HouseCanary.com [Електронний ресурс] – Режим доступу до ресурсу: <https://www.housecanary.com/>
8. Intelligent data analysis using artificial neural networks for decision making in the education domain / P. Radiuk, O. Mazurets, T. Skrypnyk, O. Moroz // Вісник Хмельницького національного університету. Технічні науки. – 2021. – №6. – С. 111-114.
9. Keras [Електронний ресурс]. – Режим доступу: <https://keras.io/>
10. Lun.ua [Електронний ресурс] – Режим доступу до ресурсу: <https://lun.ua/>
11. Machine Learning & Data Science Landscape by Christina Voskoglou, Mark Wilcox, Stijn Schuermans – VisionMobile 2017, 46 p.
12. Matplotlib [Електронний ресурс]. – Режим доступу: <https://matplotlib.org/>

13. Moroz, O.H. Problem of Constructing the GMDH Neural Networks with Active Neurons / O. H. Moroz, Y. M. Linder // Системи керування та комп'ютери. – 2022. – №2. – С. 33-42.

14. NanoNets: How to use Deep Learning when you have Limited Data [Електронний ресурс]. – Режим доступу: <https://medium.com/nanonets/nanonets-how-to-use-deep-learning-when-you-have-limited-data-f68c0b512cab>

15. Nest [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.nestjs.com/>

16. Node.JS [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/uk/docs>

17. Pandas [Електронний ресурс]. – Режим доступу: <https://pandas.pydata.org/>

18. Price.ua [Електронний ресурс] – Режим доступу до ресурсу: <https://price.ua/ua>

19. Python [Електронний ресурс]. – Режим доступу: <https://www.python.org/>

20. Python for Data Analysis, 2E By Wes McKinney – O'Reilly Media 2016, 550 p.

21. Redfin.com [Електронний ресурс] – Режим доступу до ресурсу: <https://www.redfin.com/>

22. Singh, A.; Sharma, A.; Dubey, G. Big data analytics predicting real estate prices. Int. J. Syst. Assur. Eng. Manag. 2020.

23. TensorFlow [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/>

24. The Multi-Channel Neural Network Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/the-multi-channel-neural-network-26551bdfab6c>.

25. Trulia.com [Електронний ресурс] – Режим доступу до ресурсу: <https://www.trulia.com/>

26. Zillow.com [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zillow.com/>

27. Воронін В.О. Дослідження ринку нерухомості. Проблеми, тенденції, прогнозування / В.О. Воронін, Е. В. Лянце, М. М. Мамчин // Вісник Національного університету «Львівська політехніка», 2010. – No 690. – С. 540–552.

28. Кравчук, О.А.М.А.В. Штучні нейронні мережі: визначення та сфери застосування / О. А. Кравчук // Сучасні досягнення у науці та освіті. – 2019. – С. 167-171.

29. Ринок нерухомості України. [Електронний ресурс]. – Режим доступу: <https://minfin.com.ua/ua/realty/>

30. Штучний інтелект з Python [Текст] : навч. посіб. / Р. І. Ліщук [та ін.] ; Уман. нац. ун-т садівництва. - Умань : Сочінський М. М. [вид.], 2021. - 79 с.

ДОДАТКИ

Додаток А

Програмний код

```

#Model.py
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

Data = pd.read_csv('data_house.csv')
Data.head(5).T

Data.info()
Data.describe().transpose()

fig = plt.figure(figsize=(10,7))
fig.add_subplot(2,1,1)
sns.histplot(Data['price'],kde=True)
fig.add_subplot(2,1,2)
sns.boxplot(Data['price'])
plt.tight_layout()

fig = plt.figure(figsize=(16,5))
fig.add_subplot(2,2,1)
sns.scatterplot(Data,x='sqft_above', y='price')
fig.add_subplot(2,2,2)
sns.scatterplot(Data,x='sqft_lot',y='price')
fig.add_subplot(2,2,3)
sns.scatterplot(Data,x='sqft_living',y='price')
fig.add_subplot(2,2,4)
sns.scatterplot(Data,x='sqft_basement',y='price')

fig = plt.figure(figsize=(15,7))
fig.add_subplot(2,2,1)
sns.countplot(x=Data['bedrooms'])
fig.add_subplot(2,2,2)
sns.countplot(x=Data['floors'])
fig.add_subplot(2,2,3)
sns.countplot(x=Data['bathrooms'])
fig.add_subplot(2,2,4)
sns.countplot(x=Data['grade'])
plt.tight_layout()

Data['date'] = pd.to_datetime(Data['date'])
Data['month'] = Data['date'].apply(lambda date:date.month)
Data['year'] = Data['date'].apply(lambda date:date.year)

fig = plt.figure(figsize=(16,5))
fig.add_subplot(1,2,1)
Data.groupby('month').mean()['price'].plot()
fig.add_subplot(1,2,2)
Data.groupby('year').mean()['price'].plot()

Data.isnull().sum()

Data = Data.drop('date',axis=1)
Data = Data.drop('id',axis=1)
Data = Data.drop('zipcode',axis=1)

X = Data.drop('price',axis =1).values
y = Data['price'].values
from sklearn.model_selection import train_test_split

```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=101)

from sklearn.preprocessing import StandardScaler
s_scaler = StandardScaler()
X_train = s_scaler.fit_transform(X_train.astype(float))
X_test = s_scaler.transform(X_test.astype(float))

from joblib import dump
dump(s_scaler, 's_scaler.bin', compress=True)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation
from tensorflow.keras.optimizers import Adam

model = Sequential()
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))
model.add(Dense(19,activation='relu'))
model.add(Dense(1))
model.compile(optimizer='Adam',loss='mse')

model.fit(x=X_train,y=y_train,
          validation_data=(X_test,y_test),
          batch_size=128,epochs=400)
model.summary()

loss_df = pd.DataFrame(model.history.history)
loss_df.plot(figsize=(12,8))

y_pred = model.predict(X_test)
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('      ', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('VarScore:',metrics.explained_variance_score(y_test,y_pred))
fig = plt.figure(figsize=(10,5))
plt.scatter(y_test,y_pred)
plt.plot(y_test,y_test,'r')
model.save('Pred_AI')

import sys
import tensorflow as tf
import pandas as pd
from joblib import load
import os

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

#Load_Model.py
model = tf.keras.models.load_model("D:\JS\diploma\api\src\prediction\python\Pred_AI")

house_df =pd.DataFrame({
    'bedrooms':int(sys.argv[1]),
    'bathrooms':float(sys.argv[2]),
    'sqft_living':int(sys.argv[3]),
    'sqft_lot':int(sys.argv[4]),
    'floors':float(sys.argv[5]),
    'waterfront':int(sys.argv[6]),
    'view':int(sys.argv[7]),
    'condition':int(sys.argv[8]),

```

```

        'grade':int(sys.argv[9]),
        'sqft_above':int(sys.argv[10]),
        'sqft_basement':int(sys.argv[11]),
        'yr_built':int(sys.argv[12]),
        'yr_renovated':int(sys.argv[13]),
        'lat':float(sys.argv[14]),
        'long':float(sys.argv[15]),
        'sqft_living15':int(sys.argv[16]),
        'sqft_lot15':int(sys.argv[17]),
        'month':int(sys.argv[18]),
        'year':int(sys.argv[19])
    },index=[0])

s_scaler=load("D:\JS\diploma\api\src\prediction\python\s_scaler.bin")
predict_data = s_scaler.transform(house_df.values.astype(float))

predicted_price = model.predict(predict_data,verbose=0)
print(predicted_price[0][0],flush=True)

//Pred_service.ts
import {Injectable} from '@nestjs/common';
import {spawnSync} from 'child_process'
@Injectable()
export class PredictionService {
    async getPrediction(data: any) {
        const pyProcess = await spawnSync('C:/Users/andri/anaconda3/python',
            ['./src/prediction/python/python.py',
                data.bedrooms,
                data.bathrooms,
                data.sqft_living,
                data.sqft_lot,
                data.floors,
                data.waterfront,
                data.view,
                data.condition,
                data.grade,
                data.sqft_above,
                data.sqft_basement,
                data.yr_built,
                data.yr_renovated,
                data.lat,
                data.long,
                data.sqft_living15,
                data.sqft_lot15,
                data.month,
                data.year,
            ])
        return {price: pyProcess.stdout.toString(), err: pyProcess.stderr.toString() ||
null}
    }
}

//Pred_controller.ts
import {Body, Controller, Post} from '@nestjs/common';
import { PredictionService } from './prediction.service';
import {MakePredictionDto} from "./dto/makePrediction.dto";

@Controller('prediction')
export class PredictionController {
    constructor(private readonly predictionService: PredictionService) {}

    @Post('make')
    async makePrediction(@Body() makePredictionDto:MakePredictionDto){
        return this.predictionService.getPrediction(makePredictionDto)
    }
}

```

```

    }
  }
}
//Pred_module.ts
import { Module } from '@nestjs/common';
import { PredictionService } from './prediction.service';
import { PredictionController } from './prediction.controller';

@Module({
  controllers: [PredictionController],
  providers: [PredictionService]
})
export class PredictionModule {}
//App_module.ts
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { PredictionModule } from './prediction/prediction.module';

@Module({
  imports: [PredictionModule],
  controllers: [AppController],
  providers: [AppService],
})
export class AppModule {}
//main.ts
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create(AppModule, {cors: true});
  await app.listen(3000);
}

bootstrap();
//prediction.reducer.ts
import {createAsyncThunk, createSlice} from "@reduxjs/toolkit";
import {predictionAPI} from "../api";

type InitialState = {
  price: string | null,
  pending: boolean,
  error: any
}
const initialState: InitialState = {
  price: null,
  pending: false,
  error: null
}
export const makePrediction = createAsyncThunk(
  'prediction/makePrediction',
  async (data: any, thunkAPI) => {
    try {
      const res = await predictionAPI.makePrediction(data)
      return thunkAPI.fulfillWithValue(res.price)
    } catch (e: any) {
      return thunkAPI.rejectWithValue(e.message)
    }
  }
)
export const predictionSlice = createSlice({
  name: 'prediction',
  initialState,
  reducers: {

```

```

        clearPrice: (state) => {
            state.price = null
        }
    },
    extraReducers: (builder) => {
        builder
            .addCase(makePrediction.fulfilled, (state, action) => {
                debugger
                state.price = action.payload
                state.pending = false
                state.error = null
            })
            .addCase(makePrediction.pending, (state) => {
                debugger
                state.pending = true
                state.error = null
            })
            .addCase(makePrediction.rejected, (state, action) => {
                debugger
                state.pending = false
                state.error = action.payload
            })
    }
})
export const {clearPrice} = predictionSlice.actions

export default predictionSlice.reducer

//prediction.selector.ts
import {RootState} from "./store.ts";

export const selectPrice = (state: RootState) => state.prediction.price
export const selectPending = (state: RootState) => state.prediction.pending
export const selectError = (state: RootState) => state.prediction.error

//store.ts
import { configureStore } from '@reduxjs/toolkit'
import { useDispatch, useSelector } from 'react-redux'
import type { TypedUseSelectorHook } from 'react-redux'
import predictionReducer from "./prediction.reducer.ts";

export const store = configureStore({
    reducer: {
        prediction: predictionReducer
    },
})

export type RootState = ReturnType<typeof store.getState>
export type AppDispatch = typeof store.dispatch

export const useAppDispatch: () => AppDispatch = useDispatch
export const useAppSelector: TypedUseSelectorHook<RootState> = useSelector

//House.form.tsx
import {FC} from 'react';
import {Button, Col, Form, InputNumber, Row} from 'antd';
import {useAppDispatch} from "../redux/store.ts";
import {makePrediction} from "../redux/prediction.reducer.ts";

type Props = {}
const HouseForm: FC<Props> = () => {
    const dispatch = useAppDispatch()
    const onFinish = (values: any) => {
        values = {...values, year: 2014, month: 1}
    }
}

```

```

    dispatch(makePrediction(values))
  };

const onFinishFailed = (errorInfo: any) => {
  console.log('Failed:', errorInfo);
};

return (
  <Form
    layout='vertical'
    name="basic"
    labelCol={{span: 0}}
    wrapperCol={{span: 0}}
    style={{width: "100%", margin: "0 auto"}}
    initialValues={{remember: true}}
    onFinish={onFinish}
    onFinishFailed={onFinishFailed}
    autoComplete="off"
  >
    <Row gutter={20} justify={"center"}>
      <Col>
        <Form.Item
          label="Bedrooms"
          name="bedrooms"
          initialValue={3}
          rules={[{required: true, message: 'Required!'}]}
        >
          <InputNumber/>
        </Form.Item>

        <Form.Item
          label="Bathrooms"
          name="bathrooms"
          initialValue={1.0}
          rules={[{required: true, message: 'Required!'}]}
        >
          <InputNumber step='0.1' />
        </Form.Item>

        <Form.Item
          label="Sqft_living"
          name="sqft_living"
          initialValue={1180}
          rules={[{required: true, message: 'Required!'}]}
        >
          <InputNumber/>
        </Form.Item>

        <Form.Item
          label="Sqft_lot"
          name="sqft_lot"
          initialValue={5650}
          rules={[{required: true, message: 'Required!'}]}
        >
          <InputNumber/>
        </Form.Item>

        <Form.Item
          label="Floors"
          name="floors"
          initialValue={1.0}
          rules={[{required: true, message: 'Required!'}]}
        >
          <InputNumber step='0.1' />

```

```

</Form.Item>
<Form.Item
  label="Waterfront"
  name="waterfront"
  initialValue={0}
  rules={[{required: true, message: 'Required!'}]}
>
  <InputNumber/>
</Form.Item>
</Col>
<Col>
  <Form.Item
    label="View"
    name="view"
    initialValue={0}
    rules={[{required: true, message: 'Required!'}]}
  >
    <InputNumber/>
  </Form.Item>

  <Form.Item
    label="Condition"
    name="condition"
    initialValue={3}
    rules={[{required: true, message: 'Required!'}]}
  >
    <InputNumber/>
  </Form.Item>

  <Form.Item
    label="Grade"
    name="grade"
    initialValue={7}
    rules={[{required: true, message: 'Required!'}]}
  >
    <InputNumber/>
  </Form.Item>

  <Form.Item
    label="Sqft_above"
    name="sqft_above"
    initialValue={1180}
    rules={[{required: true, message: 'Required!'}]}
  >
    <InputNumber/>
  </Form.Item>

  <Form.Item
    label="Sqft_basement"
    name="sqft_basement"
    initialValue={0}
    rules={[{required: true, message: 'Required!'}]}
  >
    <InputNumber/>
  </Form.Item>

  <Form.Item
    label="Yr_built"
    name="yr_built"
    initialValue={1955}
    rules={[{required: true, message: 'Required!'}]}
  >
    <InputNumber/>

```

```

        </Form.Item>
      </Col>
    <Col>
      <Form.Item
        label="Yr_renovated"
        name="yr_renovated"
        initialValue={0}
        rules={[{required: true, message: 'Required!'}]}
      >
        <InputNumber/>
      </Form.Item>

      <Form.Item
        label="Lat"
        name="lat"
        initialValue={47.5112}
        rules={[{required: true, message: 'Required!'}]}
      >
        <InputNumber/>
      </Form.Item>

      <Form.Item
        label="Long"
        name="long"
        initialValue={-122.257}
        rules={[{required: true, message: 'Required!'}]}
      >
        <InputNumber/>
      </Form.Item>

      <Form.Item
        label="Sqft_living15"
        name="sqft_living15"
        initialValue={1340}
        rules={[{required: true, message: 'Required!'}]}
      >
        <InputNumber/>
      </Form.Item>

      <Form.Item
        label="Sqft_lot15"
        name="sqft_lot15"
        initialValue={5650}
        rules={[{required: true, message: 'Required!'}]}
      >
        <InputNumber/>
      </Form.Item>
    </Col>
  </Row>
  <Row justify={"center"}>
    <Form.Item>
      <Button type="primary" htmlType="submit">
        Predict price
      </Button>
    </Form.Item>
  </Row>
</Form>
);
}
export default HouseForm;

//PredictionResult.tsx

```

```

import {useAppSelector} from "../redux/store.ts";
import {selectError, selectPending, selectPrice} from
"../redux/prediction.selectors.ts";
import {Typography} from 'antd';
import Preloader from "./Preloader.tsx";

const {Title} = Typography;

const PredictionResult = () => {
  const price = useAppSelector(selectPrice)
  const pending = useAppSelector(selectPending)
  const error = useAppSelector(selectError)
  if (pending) return <Preloader/>
  return (
    <div style={{height:"50px"}}>
      {price &&
        <Title style={{textAlign: "center"}}>
          {price}$
        </Title>
      }
      {error &&
        <Title style={{textAlign: "center"}}>
          Error:{error}
        </Title>
      }
    </div>
  )
}
export default PredictionResult

//Preloader.tsx
import {Spin, Typography} from "antd";
const {Title} = Typography
const Preloader = () => {
  return (
    <div style={{height: "50px"}}>
      <Spin style={{color: "black", marginTop: '20px'}} size='large' tip='Making
prediction...'>
        <Title className='content'/>
      </Spin>
    </div>
  )
}
export default Preloader

//App.tsx
import HouseForm from "../components/HouseForm.tsx";
import './App.css'
import PredictionResult from "../components/PredictionResult.tsx";

const App = () => {

  return (
    <div style={{width: '98vw'}}>
      <HouseForm/>
      <PredictionResult/>
    </div>
  )
}
export default App

```

Додаток Б

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА НА ТЕМУ «СПОСІБ ПЕРЕДБАЧЕННЯ ЦІНИ ЖИТЛА ЗАСОБАМИ ШТУЧНОГО ІНТЕЛЕКТУ У РОЗПОДІЛЕНИХ РІЄЛТОРСЬКИХ СИСТЕМАХ »

Виконав:

Ст. групи КН-19(2)

Таланчук Андрій Олексійович

Керівник:

к.т.н, доцент кафедри КН

Пасічник Олександр Анатолієвич

Актуальність теми

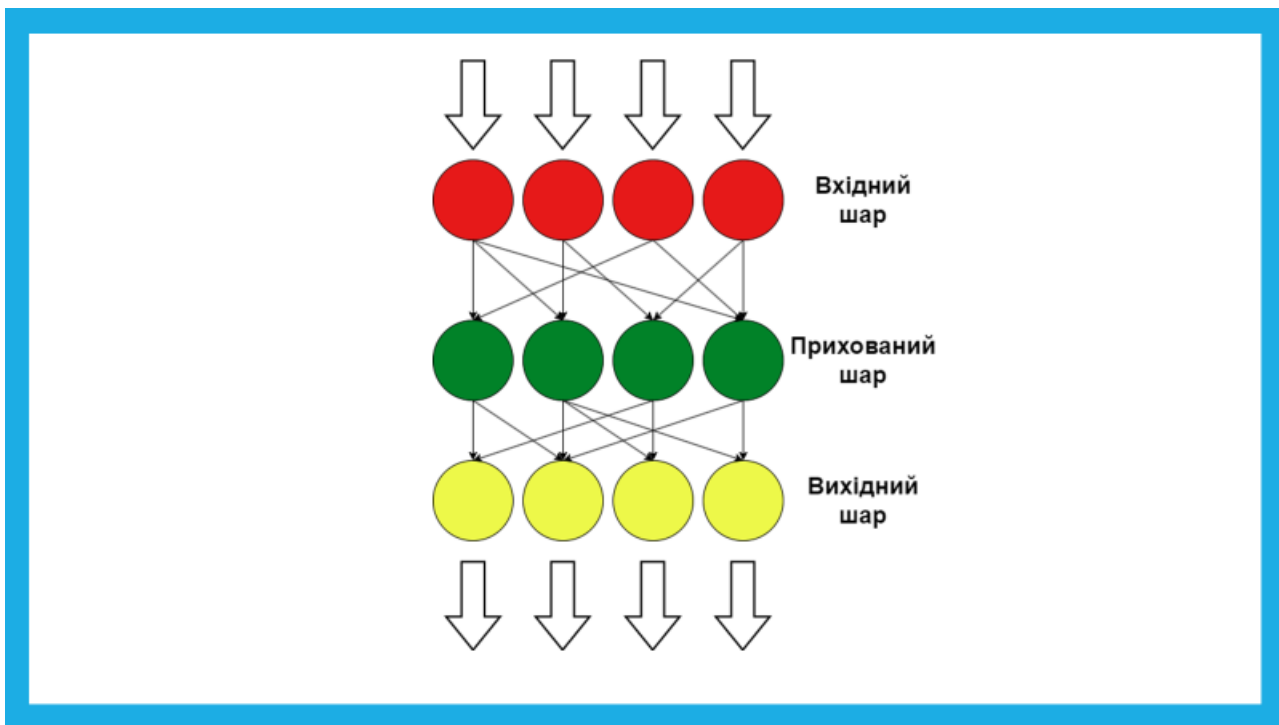
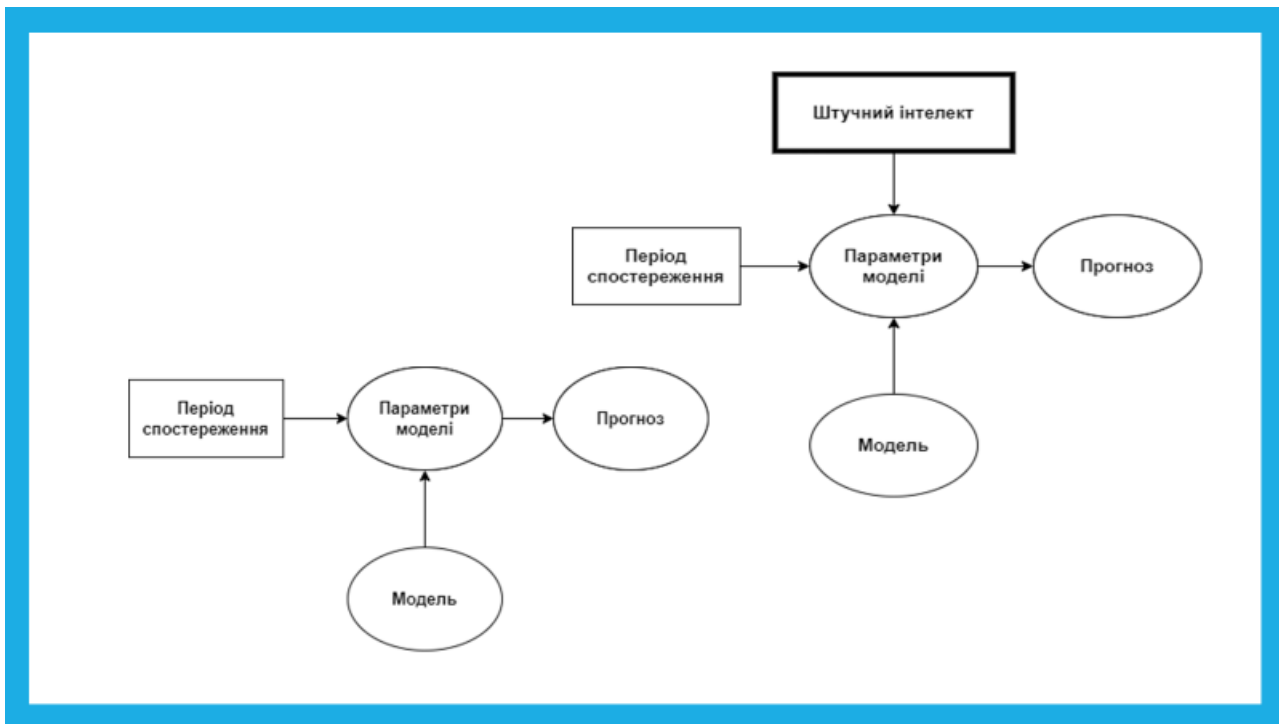
- Об'єктивний розвиток науки та технологій у поєднанні зі зростання можливостей технічних засобів визначили стрімкий розвиток методів та систем штучного інтелекту. Можливості методів штучного інтелекту, їх універсальність та ефективність обумовлюють їх широке впровадження у різноманітні сфери діяльності людини.
- Першочерговими є задачі пов'язані із прогнозування результатів діяльності, особливо коли мова йдеться про ресурсоємкі проекти із суттєвими різноманітними обмеженнями. Однією з найбільш складних та відповідальних напрямків діяльності, особливо після повномасштабного вторгнення в Україну, є побудова нового та відбудова зруйнованого житлового фонду. Величезні обсяги цієї задачі у поєднанні із об'єктивними обмеженнями, фінансовими, товарними та часовими ресурсами обумовлюють необхідність ретельного планування цієї діяльності.

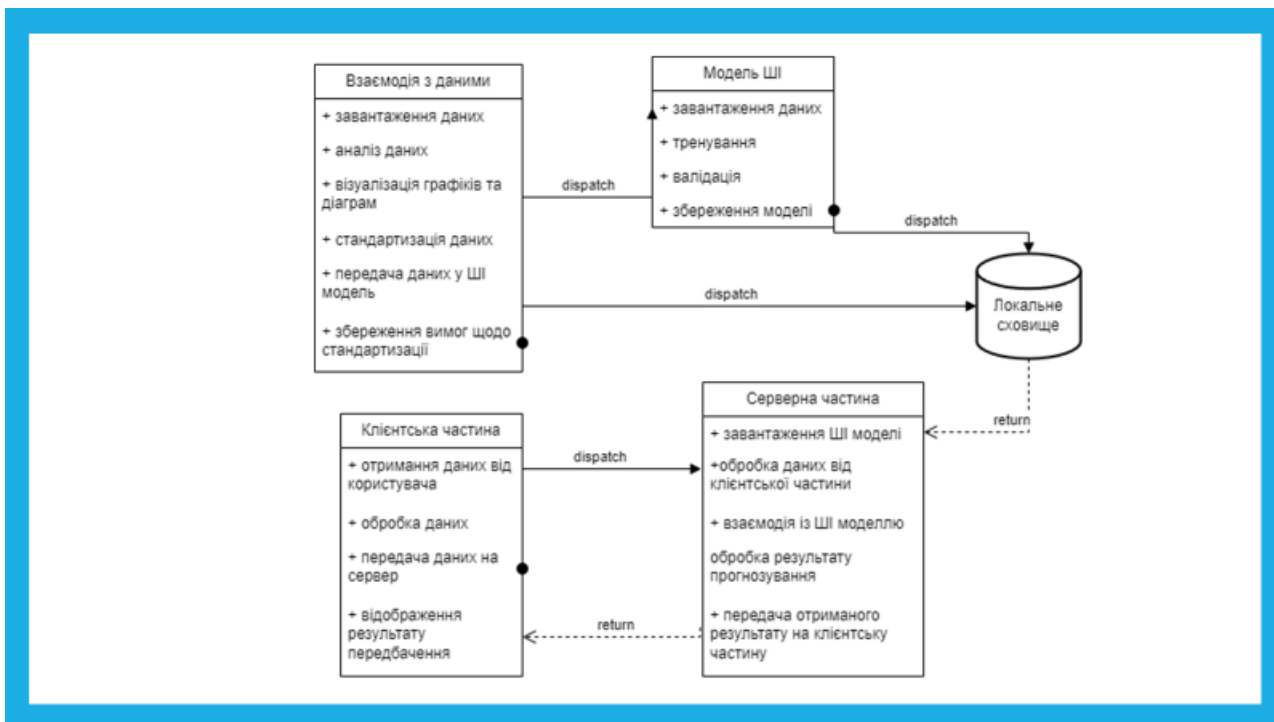
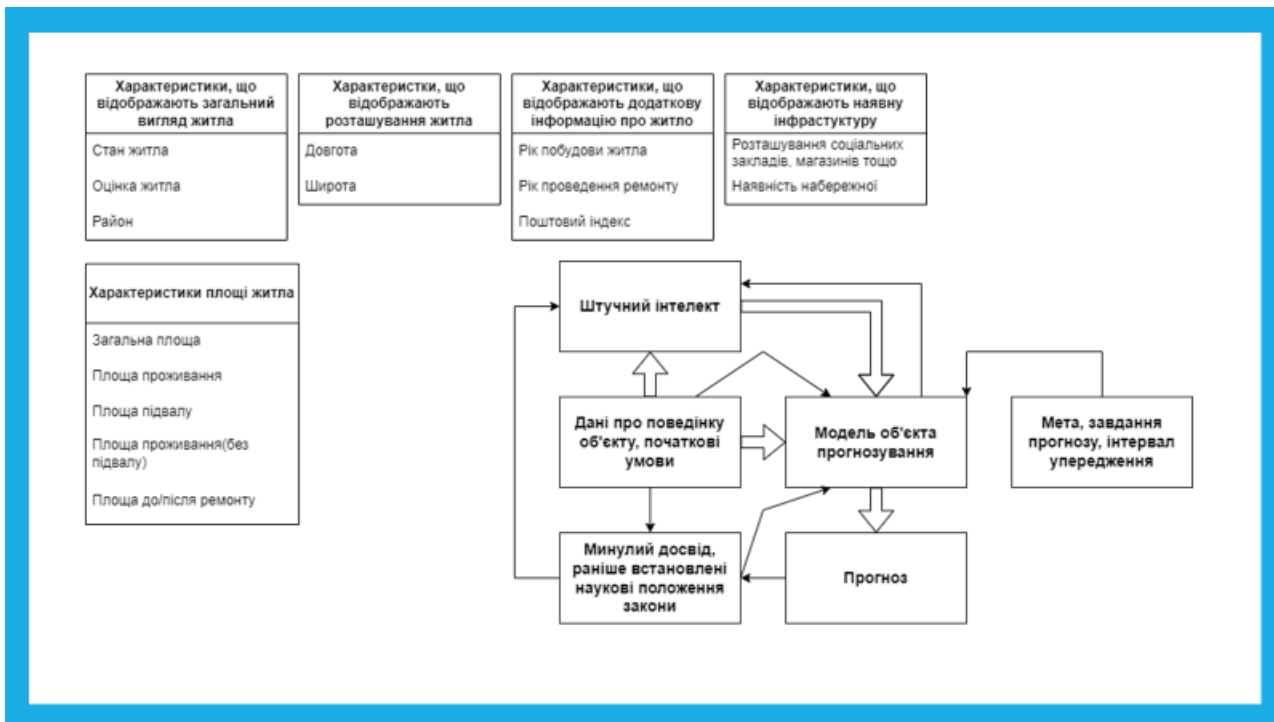
Мета і задачі

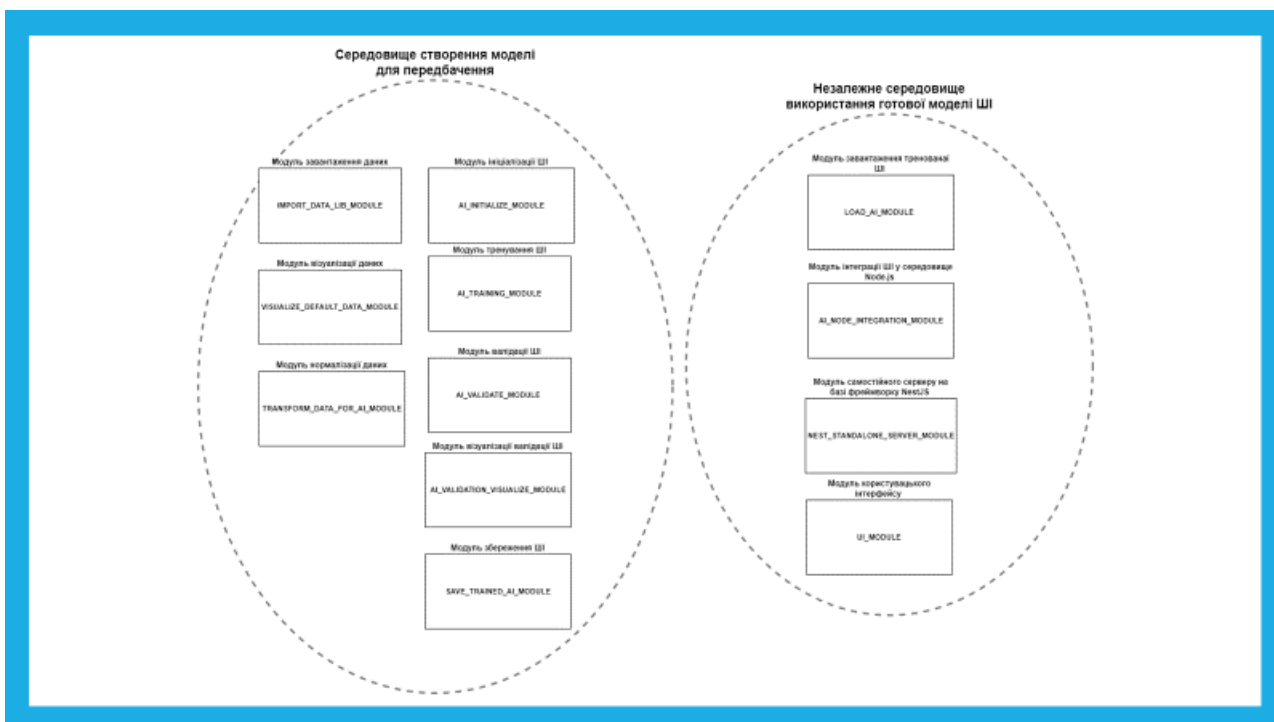
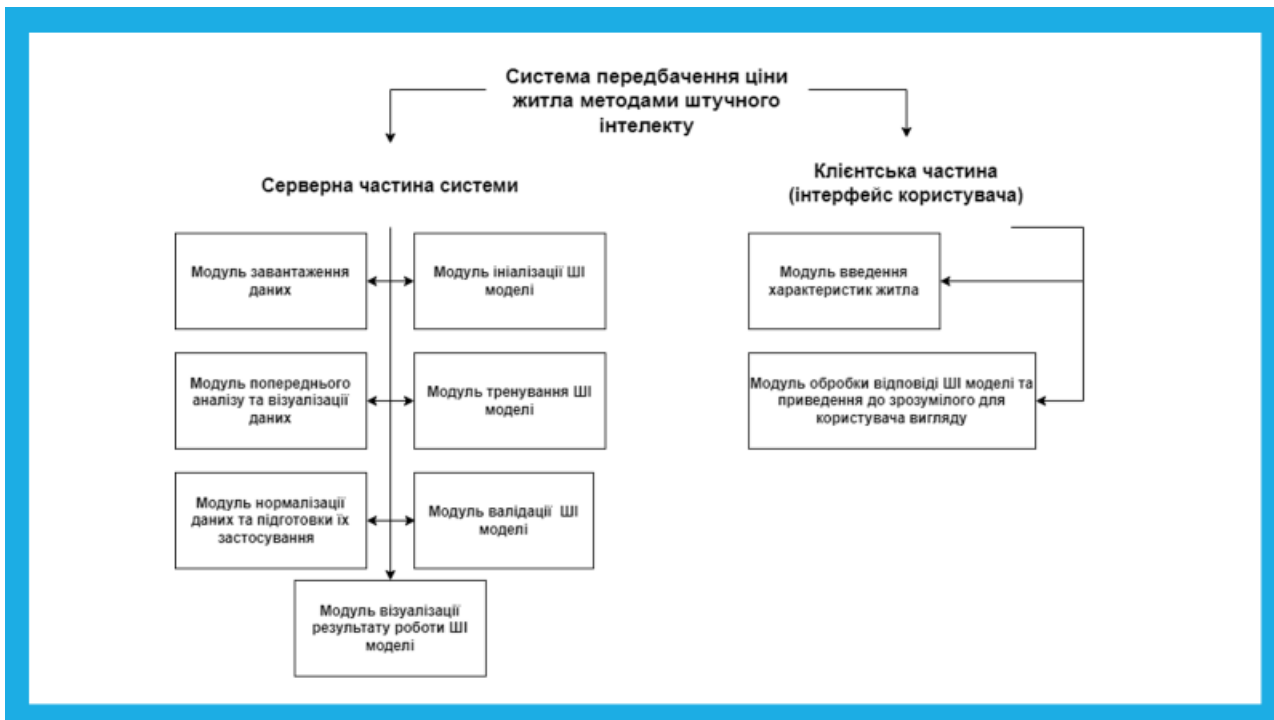
- Мета кваліфікаційної роботи полягає в створенні способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.
- Для досягнення поставленої мети визначені наступні задачі дослідження:
 - аналіз сучасних способів передбачення ціни житла ріелторських системах;
 - аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем;
 - реалізувати інформаційну технологію способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах;
 - провести експериментальне тестування інформаційної технології.

Об'єкт та предмет дослідження

- **Об'єкт дослідження** – процес побудови способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.
- **Предмет дослідження** – моделі, алгоритми та засоби для створення способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.

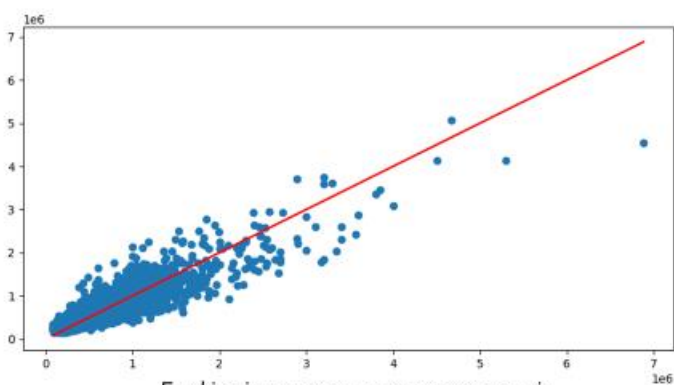






	0	1	2	3	4
id	7129300520	6414100192	5631500400	2487200875	1954400510
date	20141013T000000	20141209T000000	20150225T000000	20141209T000000	20150218T000000
price	221900.0	538000.0	180000.0	604000.0	510000.0
bedrooms	3	3	2	4	3
bathrooms	1.0	2.25	1.0	3.0	2.0
sqft_living	1180	2570	770	1960	1680
sqft_lot	5650	7242	10000	5000	8080
floors	1.0	2.0	1.0	1.0	1.0
waterfront	0	0	0	0	0
view	0	0	0	0	0
condition	3	3	3	5	3
grade	7	7	6	7	8
sqft_above	1180	2170	770	1050	1680
sqft_basement	0	400	0	910	0
yr_built	1955	1951	1933	1965	1987
yr_renovated	0	1991	0	0	0
zipcode	98178	98125	98028	98136	98074
lat	47.5112	47.721	47.7379	47.5208	47.6168
long	-122.257	-122.319	-122.233	-122.393	-122.045
sqft_living15	1340	1690	2720	1360	1800
sqft_lot15	5650	7639	8062	5000	7503

Вигляд набору даних, що використовується для тренування і валідації моделі ШІ



Графік відношення прогнозованих цін до фактичних

Bedrooms

View

Yr_renovated

Bathrooms

Condition

Lat

Sqft_living

Grade

Long

Sqft_lot

Sqft_above

Sqft_living15

Floors

Sqft_basement

Sqft_lot15

Waterfront

Yr_built

274608.44 \$

Результат передбачення за заданими користувачем характеристиками

ВИСНОВОК

- В кваліфікаційній роботі створено спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах.
- Для досягнення поставленої мети було вирішено такі задачі дослідження: проаналізовано сучасні способи передбачення ціни житла у ріелторських системах; виконано аналіз методів та засобів штучного інтелекту в задачах прогнозування та сучасних розподілених систем; реалізована інформаційна технологія способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах; проведено експериментальне тестування інформаційної технології.
- Виконана валідація моделі підтвердила її коректність, похибки перевірки на тестовій виборці склали: ~ 102507 – для середньої абсолютної помилки, ~ 26652574653 – для середньої квадратичної помилки, ~ 163256 – для кореневого середньоквадратичного відхилення, точність моделі становить $\sim 80\%$.

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 10%**

ID: 114258 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-05-29 Автора: А.О. Таланчук Керівники: О.А.Пасічник Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	52665	796	2048 (4%)	28 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

ID перевірки:
1015305052

Дата перевірки:
29.05.2023 20:30:33 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
29.05.2023 20:34:13 EEST

ID користувача:
100005671

Назва документа: КН-19-2 Таланчук

Кількість сторінок: 60 Кількість слів: 8098 Кількість символів: 62944 Розмір файлу: 1.89 MB ID файлу: 1014976520

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.92% Схожість

Найбільша схожість: 1.59% з джерелом з Бібліотеки (ID файлу: 1014976252)

1.52% Джерела з Інтернету

135

Сторінка 62

1.78% Джерела з Бібліотеки

105

Сторінка 63

0% Цитат

Не знайдено жодних цитат

Посилання

1

Сторінка 63

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

Підозріле форматування

22
сторінки

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

Автор: Таланчук Андрій Олексійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент Пасічник Олександр Анатолійович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 2 %.

2) за програмою UNICHECK виявлені 1.92%.; Найбільша схожість: Найбільша схожість: 1.59% з джерелом з Бібліотеки (ID файлу: 1014976252), яке містить матеріали огляду предметної області; інші схожості є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.

збігів/ідентичності/схожості, складає 2 % і 1.92% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН

Олександр ПАСІЧНИК

Олександр МАЗУРЕЦЬ

Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента гр. КН-19-2 Таланчука Андрія Олексійовича

за темою Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

1. Актуальність теми

Об'єктивний розвиток науки та технологій у поєднанні зі зростання можливостей технічних засобів визначили стрімкий розвиток методів та систем штучного інтелекту. Можливості методів штучного інтелекту, їх універсальність та ефективність обумовлюють їх широке впровадження у різноманітні сфери діяльності людини. Першочерговими є задачі пов'язані із прогнозування результатів діяльності, особливо коли мова йдеться про ресурсомкі проекти із суттєвими різноманітними обмеженнями. Однією з найбільш складних та відповідальних напрямків діяльності, особливо після повномасштабного вторгнення в Україну, є побудова нового та відбудова зруйнованого житлового фонду. Значні обсяги капіталовкладень у поєднанні із об'єктивно обмеженими ресурсами обумовлюють необхідність ретельного планування цієї діяльності.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є розробка способу передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах та розробка інформаційної системи реалізації вказаного способу. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, а саме засоби штучного інтелекту, дозволило розробити ці інформаційних технологій. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Таланчук Андрій Олексійович проявив себе кваліфікованим фахівцем та дисциплінованим студентом.

вчасно виконуючи поставлені етапи дослідження. Під час написання пояснювальної записки й при розробці прикладного програмного забезпечення проявив достатні для одержання успішного результату загальні та фахові компетентності та продемонстрував досягнення програмних результатів навчання за напрямком «Комп'ютерні науки».

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідним інструментарієм, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та розроблено програмне забезпечення для валідації та верифікації запропонованого метода.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі спосіб передбачення ціни житла засобами штучного інтелекту у розподілених релторських системах та його програмна реалізація може бути використана фахівцями сфери продажу нерухомості на первинному та вторинному ринках житла, а також забудовниками та інвесторами.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Керівник



к.т.н., доцент Олександр ПАСІЧНИК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-2 Таланчука Андрія Олексійовича
за темою: Спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах

1. Актуальність обраної теми

Сучасний історичний етап характеризується значними руйнуваннями, великою кількістю біженців та внутрішніх переселенців. Це визначає значну потребу у житловому будівництві вже зараз та у найближчому майбутньому. Будівництво відноситься до сфери капітальних вкладень є ресурсомким. Можливість формування якомога точніших оцінок ціни житла сприяє кращому плануванню відбудови.

2. Повнота розкриття мети та завдань роботи

Мета роботи розкрита повністю. Ісі поставлені завдання вирішені

3. Зміст кожного розділу роботи

В першому розділі виконано аналіз предметної області включно із сучасними способами передбачення ціни житла та застосування для цього методів штучного інтелекту. Сформульовано мету та визначено задачі.

В другому розділі реалізовано спосіб передбачення ціни житла із використанням нейронних мереж, спроектовано структуру інформаційної системи з деталізацією функціональної будови.

В третьому розділі виконано програмну реалізацію способу та виконано його валідацію.

4. Оцінка розробленої інформаційної системи, її практична цінність
Розроблений у роботі спосіб передбачення ціни житла засобами штучного інтелекту у розподілених ріелторських системах та його програмна реалізація може бути використаний у сфері нерухомості забудовниками, ріелторами та інвесторами.

5. Якість оформлення кваліфікаційної роботи бакалавра

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

6. Недоліки кваліфікаційної роботи бакалавра

З роботи не зрозуміло, яким чином її результати можуть бути використані для вітчизняного ринку

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Рецензент

Д-р. каф. ТМІТ (ХНУ)

Олег Ковалев