Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

# NLP: СТВОРЕННЯ ПАРСЕРА ДЛЯ СТРУКТУРУВАННЯ ТЕКСТУ ОГОЛОШЕНЬ НЕРУХОМОСТІ

**Текстова частина до курсової роботи**
**за спеціальністю «Інженерія програмного забезпечення» 121**

Керівник курсової роботи
асистент Смиш О.Р.
_____
(підпис)
«____» _____ 2023 р.

Виконала студентка Махиня А.О.
«____» _____ 2023 р.

Київ 2023

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри мультимедійних систем,
к.ф.-м.н.
_____ О. П. Жежерун
*(підпис)*
«____»_____2022 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**
на курсову роботу

студентці Махині Анастасії Олександрівні факультету інформатики 3 курсу
ТЕМА «NLP: створення парсера для структурування тексту оголошень нерухомості»
Зміст ТЧ до курсової роботи:

Зміст
Анотація
Вступ
1. Обробка природної мови
2. Збір оголошень про нерухомість
3. Аналіз текстів оголошень нерухомості
4. Розробка парсера
5. Використання парсера
Висновки
Список літератури
Додатки

Дата видачі «___» _____ 2022 р. Керівник _____
*(підпис)*
Завдання отримав _____
*(підпис)*

Тема: NLP: створення парсера для структурування тексту оголошень нерухомості

Календарний план виконання роботи:

| № п/п | Назва етапу | Термін виконання етапу | Примітка |
|---|---|---|---|
| 1. | Отримання теми курсової роботи | 16.07.2022 | |
| 2. | Огляд технічної літератури за темою роботи | 01.08.2022 – 26.04.2023 | |
| 3. | Аналіз сучасних методів обробки природної мови | 18.08.2022 – 05.09.2023 | |
| 4. | Збір оголошень про нерухомість | 05.09.2022 – 12.03.2023 | |
| 5. | Аналіз оголошень про нерухомість | 23.10.2022 – 13.11.2022 | |
| 6. | Програмування методів для парсеру | 13.11.2022 – 06.04.2023 | |
| 7. | Створення моделі машинного навчання | 12.02.2023 – 06.04.2023 | |
| 8. | Розробка інтерфейсу для демонстрації роботи парсера | 22.03.2023 – 11.05.2023 | |
| 9. | Написання текстової частини курсвої роботи | 06.04.2023 – 11.05.2023 | |
| 10. | Перегляд змісту роботи з керівником | 11.05.2023 | |
| 11. | Створення слайдів для доповіді та написання доповіді | 15.05.2023 | |
| 12. | Захист курсової роботи | 25.05.2023 | |

Студент _____

Керівник __ _____

“_____”_____2023 р.

# Зміст

Анотація

У роботі описано створення парсера для структурування оголошень нерухомості за допомогою сучасних можливостей обробки природної української мови, що базується на аналізі текстів оголошень та доступних засобів обробки живої мови.

Кінцевий програмний продукт здатен визначати та виділяти ключові елементи оголошень, що дає змогу полегшити подальшу обробку та аналіз даних. Він може бути використаний в різних сферах, таких як аналіз ринку нерухомості, системи рекомендацій або автоматична побудова баз даних.

# Вступ

Обробка природної мови на сьогодні одна з найпопулярніших галузей математичної лінгвістики. У сучасному світі постійно зростає обсяг неструктурованої текстової інформації. Створення застосунків, які вміють взаємодіяти з текстом поданим природною мовою, набувають все більшого попиту, оскільки вони поліпшують процес пошуку та аналізу потрібної інформації.

Проте наразі немає готового рішення, для структурування тексту оголошень про нерухомість.

Мета дослідження полягає у тому, щоб проаналізувати наявні рішення обробки природньої української мови й розглянути можливість використання такого рішення для створення парсера для структурування текстів оголошень нерухомості.

Завдання роботи ґрунтується на створенні парсера, що дає змогу структурувати оголошення нерухомості, виокремлюючи головні характеристики квартири.

Робота складається з п’яти розділів.

Перший розділ дає змогу ознайомитись з обробкою природних мов та їхніми можливостями.

У другому розділі проводиться аналіз методів та стратегій, що застосовуються для ефективного збору оголошень про нерухомість з різних джерел.

Третій розділ присвячено аналізу особливостей оголошень про нерухомість. Визначаються основні складові елементи, які зазвичай містяться у текстах.

Третій розділ освітлює наявні на сьогодні рішення для української мови. Показано їхні можливості та пояснено визначення найкращого рішення для цієї роботи.

Четвертий розділ присвячений процесу створення програмного інструменту - парсера, який призначений для отримання та структурування інформації з текстових джерел, а саме оголошень про нерухомість.

П’ятий розділ є завершальним та демонструє приклад використання розробленого парсера для отримання інформації з оголошень про нерухомість, а саме: модель для оцінки вартості нерухомості.

# 1 Обробка природньої мови

## 1.1 Ознайомлення з концепцією та метою обробки природньої мови

Природна мова – це засіб комунікації, який використовують люди у повсякденному житті для вираження думок, ідей, почуттів та обміну інформації один між одним. Вона складається зі слів, фраз і речень, які можуть бути висловлені усним або письмовим шляхом. Природна мова є складною, різноманітною та неоднозначною, її розуміння багато в чому залежить від контексту та культурних особливостей, тому обробка природної є важливою для розвитку штучного інтелекту.

Обробка природної мови (Natural Language Processing або NLP) – це галузь в комп'ютерних науках та лінгвістиці, яка спрямована на розробку комп'ютерних алгоритмів та програм для аналізу та розуміння природної мови. Загалом, метою обробки природної мови є забезпечити комп'ютери здатністю розуміти, інтерпретувати та відтворювати природню мову.

Галузі застосування NLP охоплюють машинний переклад, аналіз емоційного забарвлення, класифікацію та генерацію тексту. У сучасному світі великої кількості інформації обробка природної мови дає змогу збільшити продуктивність та ефективність роботи з великими обсягами текстових даних.

Розвиток NLP відкриває багато нових можливостей у різних галузях, як-от медицина, маркетинг, фінанси та право. Наприклад, компанії використовують NLP для аналізу соціальних мереж, щоб зрозуміти реакцію клієнтів на продукт. У правовій сфері NLP може бути використано для автоматичного аналізу та класифікації юридичних документів, що може значно збільшити ефективність роботи юристів. Також застосування NLP можливе у сфері підтримки користувачів: створення ботів або генерація автоматичних відповідей, щоб покращити швидкість та якість обслуговування.

Наразі найбільше можливостей обробки природної мови доступно для англійської мови, оскільки вона має статус міжнародної й відома майже у всьому світі. Для обробки української мови доступна обмежена кількість можливостей, що ускладнює роботу.

## 1.2 Огляд основних понять NLP

Далі розглянемо деякі технології, пов’язані з NLP, які присутні в сучасних аналізаторах і використано в цій роботі.

Токенізація тексту — це метод поділу документа на менші частини, які зазвичай називають токенами. Кожен токен має відповідне значення. Ці токени можна класифікувати як слово, речення, тема, фраза або будь-яка інформаційна одиниця залежно від завдання аналізу тексту.[1] У контексті NLP, токенізація тексту є важливим етапом перед аналізом тексту.

Розмічування частин мови (Part-of-speech tagging (POS tagging або PoS tagging)) – це процес розмітки кожного слова в тексті певною частиною мови, ґрунтуючись не тільки на його визначенні, а й контексті, тобто його зв’язку з суміжними та спорідненими словами в тексті. POS tagging в українській мові може бути важчим, оскільки українська мова має складну морфологію і граматику. Наприклад, дієслово «писати» може мати такі форми як «пишу», «писала», «писатимуть» та інші.

Стемінг (Stemming) – це процес скорочення слова відкиданням допоміжних частин, як-от закінчення чи суфікс. Результат стемінгу іноді може бути схожий на визначення кореня слова, але слово після обробки алгоритмом стемінгу може відрізнятися від морфологічного кореня слова тому, що його алгоритми базуються на інших принципах. Це скорочує кількість унікальних слів у тексті. Наприклад, слово "бігати", "бігають", "бігав" мають спільний стем "біг", що дає змогу їх обробляти як одне слово.

Лематизація (Lemmatization) – це процес зведення слова до його базової форми, яка відображає його лексичне значення в мові й називається лема. Наприклад, слова «бігла», «біжу», «бігали» при лематизації зводитимуться до леми «бігти». Так само як і стемінг, лематизація зменшує обсяг даних для аналізу тексту, проте процес лематизації складніший, ніж стемінг, бо працює на основі словника й потребує більш глибокого розуміння граматичних правил.

Розпізнавання іменованих сутностей (Named entity (N.E.) recognition) – це процес виявлення іменованих сутностей, а саме імен людей, назви міст, вулиць, дати та інших. Для NER необхідні певні знання про мову: граматичні правила та знання про іменовані сутності.

# 2 Збір оголошень про нерухомість

## 2.1 Web scrapping

Для створення парсера, який аналізуватиме та структуруватиме тексти оголошень нерухомості, зібрано дані, на основі яких проведено подальший аналіз та тестування. Оскільки в сучасному світі Інтернет є найбільш доступним джерелом інформації та існує багато сайтів створених для розміщення оголошень нерухомості, для отримання даних використано web scrapping.

Web scrapping – автоматизований процес вилучення даних з вебсайтів за допомогою програмних інструментів. Дані збираються та форматуються відповідно до конкретних потреб користувача. Метою web scrapping є отримання потрібної інформації з вебсторінок для подальшого аналізу та використання. Ця технологія дає можливість компаніям і приватним особам збирати дані у великих обсягах і отримувати інформацію, яку було б важко або неможливо отримати в інший спосіб.[2] У загальному, web scraping є потужним інструментом для отримання корисної інформації з вебсторінок.

## 2.2 Вибір сайтів для збору оголошень

Важливою частину збору оголошень нерухомості є підбір сайтів, оскільки на цьому етапі сформовано основну базу інформації, яку оброблюватиме парсер. Оскільки парсер створено для обробки природної української мови, вирішено обрати Львів та Львівську область як основний регіон для дослідження. Таке рішення обумовлене тим, що на сайтах з продажу нерухомості при пошуку оголошень в інших регіонах можна побачити оголошення написані іншою мовою.

Одним із критеріїв вибору сайтів є кількість та актуальність оголошень . На сайтах з більшою кількістю оголошень відповідно можна отримати більше даних

для детальнішого та змістовнішого аналізу. Ще важливо, щоб оголошення були актуальні, для досягнення цієї мети можна звернути увагу на вебсторінки, які часто оновлюються. Також, можна звернутися до сайтів, які спеціалізуються на продажі нерухомості в бажаній локації.

Другим критерієм є зручність форматування та доступу до даних. Деякі сайти містять багато динамічних елементів або складну структуру, в якій важко знайти потрібні CSS-селектори, що ускладнює web scraping.

Третім критерієм є доступність сайту для парсингу, оскільки не на всіх сайтах можна збирати дані без обмежень. Сайти можуть:

- блокувати IP-адреси, які надсилають багато запитів до сервера;
- використовувати CAPTCHA (Completely Automated Public Turing Test To Tell Computers and Humans Apart) – це програма, яка захищає вебсайти від вебботів, генеруючи тести, які не може пройти комп'ютер, але може пройти людина.[3];
- вимагати авторизацію.

За цими критеріями обрано три сайти, а саме:

- dom.ria.com [4];
- rieltor.ua[5];
- dim.lviv.ua[6].

## 2.3 Вибір бібліотеки для web scrapping

Python пропонує різноманітні бібліотеки, які можна використовувати для сканування Інтернету, такі як Scrapy, Beautiful Soup, Selenium.

Scrapy — це платформа для сканування вебсайтів і вилучення структурованих даних, які можна використовувати для широкого спектру корисних програм, як-от

інтелектуальний аналіз даних, обробка інформації чи історичний архів. Незважаючи на те, що Scrapy спочатку розроблено для web scraping, його також можна використовувати для вилучення даних за допомогою API (наприклад, Amazon Associates Web Services ) або як вебсканер загального призначення. [7]

Beautiful Soup – це бібліотека Python, за допомогою якої можна аналізувати HTML та XML файли. Вона будує дерево синтаксичного аналізу для проаналізованих сторінок, яке можна використовувати для вилучення даних з HTML[8].

Selenium – це вебінструмент автоматизації з відкритим вихідним кодом. Вебдрайвер Selenium має ряд функцій, які уможливлють переміщення по вебсторінках і отримування різні частини сторінок у залежності від їхніх потреб. У результаті можна отримати та впорядкувати багато даних із декількох вебсайтів, пов'язаних із запитом користувача.[8]

Beautiful Soup підходить для невеликих комплексних проєктів, Selenium оптимальне рішення для проєктів, що використовують Core JavaScript, а Scrapy найкращий вибір для великих і складних проєктів. Також, Beautiful Soup має досить низьку продуктивність порівняно з іншим бібліотеками, а найшвидше працює Scrapy, завдяки використанню асинхронних системних викликів.[8]

На основі наведеного вище порівняння визначено, що Scrapy дає кращі результати при оцінці продуктивності та функціональності різних бібліотек, тому обрано саме цей інструмент для збору оголошень про нерухомість.

## 2.4 Scrapy

Для початку потрібно налаштувати середовище обробки, встановивши Scrapy та його залежності за допомогою утиліти pip. Після успішного встановлення Scrapy, можна створити новий проєкт за допомогою команди:

*scrapy startproject*.

Далі для збору даних з вибраних сайтів потрібно створити павука (Spider) в Scrapy. Павуки (Spiders) – це класи, які визначають, як виконуватиметься сканування певного сайту (або групи сайтів), зокрема, як виконувати сканування (тобто переходити за покликаннями) і як витягувати структуровані дані з їхніх сторінок (тобто елементи сканування). Іншими словами, павуки - це місце, де ви визначаєте користувацьку поведінку для сканування й парсингу сторінок певного сайту (або, в деяких випадках, групи сайтів). [7]

Оскільки вебсторінки мають різну структуру, для кожного сайту потрібно створити окремого павука, який за допомогою URL-адреси та CSS-селекторів знайде елементи на сторінці, які містять потрібну інформацію. Після цього оголошення нерухомості збережено в текстовий файл для подальшого використання при створенні парсера.

# 3 Аналіз текстів оголошень нерухомості

## 3.1 Характеристика текстів оголошень

Оголошення нерухомості – це текстові описи житлових приміщень, які розміщують на відповідних сайтах з метою продажу. Оголошення нерухомості мають різну структуру, але є деякі типові компоненти, які зустрічаються в більшості оголошень.

Типове оголошення містить:

- заголовок оголошення;
- фото квартири;
- текстовий опис;
- інформація про вартість;
- інформація про розташування.

Оскільки власникам майна, а також ріелторам, важливо залучити більше потенційних клієнтів і виділитися серед конкурентів, текстові описи в оголошеннях, які містять багато важливої інформації (кількість кімнат, поверх і т.і.), доволі різноманітні й одразу складно виділити структуру тексту. У такому випадку може допомогти проведення кількісного аналізу. Кількісний аналіз дає змогу виявити ключові терміни, які найчастіше зустрічаються в тексті оголошень, показавши частоту використання слів у всіх оголошеннях. Завдяки цьому можна зосередитися на найбільш значущих характеристиках нерухомості і використовувати їх. Наприклад, якщо під час аналізу виявлено, що термін "поверх" зустрічається дуже часто, то варто виділити цю характеристику для структурування тексту. Для цього спершу варто здійснити лематизацію тексту, а також замінити скорочення слів на їхню повну форму, щоб привести оголошення до однакового виду.

Для демонстраціїї наведено приклад тексту оголошення нерухомості: «Продаж квартири для родини з дітьми! Всі кімнати ізольовані!3 кімнатна квартира в зданій, цегляній новобудові на 2 із 15 поверсі по вулиці Трускавецька, місто Львів.Загальна площа 83, 5 м.кв., житлова 41 м.кв., кухня 10, 2 м.кв.Висота стелі 2.7 м.Індивідуальне опалення ( 2-ох функційний котел).Будинок утеплений пінопластом 20 см.Квартира унікальна , має 2 засклені лоджії з виходом і з кухні , і з кімнати, підігрів підлоги на лоджії.2 санвузли.Тераса - 15 м.кв ( по коофіцієнту 7.3 м.кв).В квартирі зроблений підігрів підлоги, коридор, кухня, санвузли.3 кімнати ізольовані, можна зробити 2 кімнати плюс кухня-студія.Встановлені покращені ( 3 - камерні) вікна за додаткову оплату.Броньовані двері.Квартира сонячна, 2- ох сторонняВ будинку передбачений консьєрж, дитячий майданчик, парковка біля будинку.Продаж по переуступці, можна оформляти правовласність.Код об'єкту 10162.». [4]

Те саме оголошення після заміни скорочення слів на їхню повну форму та лематизації має такий вигляд: « продаж квартира родина дитина весь кімната ізольований 3 кімнатний квартира зданий цегляний новобудова 2,15 поверх вулиця трускавецький місто Львів загальний площа 83,5 м2 житловий 41 м2 кухня 10,2 м2 висота стеля 0000000000000000000000000000000 м індивідуальний опалення 2х функційний котел будинок утеплений пінопласт 20 см квартира унікальний мати 2 засклений лоджія вихід кухня кімната підігр підлога лоджія 2 санвузл тераса 15 м кв коофіцієнт 0000000000000000000000000000000 м кв квартира зроблений підігр підлога коридор кухня санвузл 3 кімната ізольований можна зробити 2 кімната плюс кухня студія встановлений покращений 3 камерний вікно додатковий оплата броньований двері квартира сонячний 2 ох сторонння будинок передбачений консьєрж дитячий майданчик парковка будинок продаж переуступка можна оформляти правовласність код об'єкт 10162».

Після заміни скорочення слів на їхню повну форму та лематизації 4284 оголошень, створено кілька файлів:

- allWords.txt – містить усі слова;
- allLemas.txt – містить усі леми;
- onlyNouns.txt – містить усі леми іменники.

Провівши кількісний аналіз лем, отримано такий результат, який наведено в таблиці.

| № | Слово | Повторюваність |
|---|---|---|
| 1 | квартира | 17040 |
| 2 | бути | 6413 |
| 3 | кімната | 5840 |
| 4 | будинок | 5250 |
| 5 | м2 | 5062 |
| 6 | 2 | 4898 |
| 7 | вулиця | 4678 |
| 8 | продаж | 4422 |
| 9 | площа | 4055 |
| 10 | стан | 3923 |

Таблиця 2.1 – 10 найбільш вживаних лем

Дані наведені в таблиці надають недостатньо інформації, тому проведено кількісний аналіз лем іменників. Отриманий результат наведено в таблиці 2.2

| № | Слово | Повторюваність |
|---|---|---|
| 1 | квартира | 17040 |
| 2 | кімната | 5838 |
| 3 | будинок | 5250 |
| 4 | м2 | 5041 |
| 5 | вулиця | 4678 |
| 6 | площа | 4055 |
| 7 | продаж | 4008 |
| 8 | стан | 3923 |
| 9 | поверх | 3704 |
| 10 | опалення | 3267 |

Таблиця 2.2 – 10 найбільш вживаних лем-іменників

З таблиці видно, що на першому місці токен «квартира», яке не надає додаткової інформації. Наступним є токен «кімната», тобто часто зазначається кількість кімнат у квартирі. На четвертому місці розташовано одиницю вимірювання площі «м2», а на шостому місці токен «площа», що вказує на те, що в оголошеннях є інформація про площу квартири. Токен «поверх» з’являється на дев’ятому місці, це дає підставу припустити, що в оголошеннях зазначають поверх чи поверховість будинку.

Таким чином, можна зробити висновок, що для парсингу оголошень про нерухомість, потрібно розробити методи, які повертатимуть інформацію про адресу, кімнатність, площу, поверх та поверховість, і додатково ціну.

## 3.2 Можливі проблеми при обробці текстів

Як було зазначено раніше, оголошення про нерухомість – це рекламні тексти, які не мають спільної чіткої структури, через це можуть виникнути додаткові труднощі.

Формат певних оголошень може містити детальну інформацію про квартиру, тоді як інших лише загальну інформацію про ціну і розташування.

Варіативність запису даних про нерухомість ускладнює розробку парсера. Наприклад, «2-кімнатна квартира», «двокімнатна квартира», «2 кімнати» - усі ці варіанти зводяться до: «в квартирі дві кімнати».

Додатково може бути наявність граматичних, лексичних та орфографічних помилок: оголошення нерухомості можуть містити помилки, які змінюють значення слова чи речення.

# 4 Розробка парсера

## 4.1 Попереднє опрацювання тексту

Перед тим, як використовувати аналізатор, спершу потрібно підготувати текст, оскільки аналізатор, який використано, не здатен зрозуміти скорочення та справитися з деякими помилками.

Першим кроком створено словник, який містить скорочення, що часто використано в оголошеннях, і повну форму слова. Наприклад, для квадратних метрів цей словник має такий вигляд:

```
'metres': {
        "м2": " м2 ",
        "кв.м.": " м2 ",
        "м.кв.": " м2 ",
        "м²": " м2 ",
        "кв.м": "м2"
    }
```

Також оскільки в деяких оголошеннях пишуть числа словами, у словник додано заміну їх на числа. Наприклад:

```
'floor':
    {
        "другий": "2",
        "другому": "2",
        "двох": "2"
    }
```

У деяких випадках для того, щоб замінити скорочення на повну форму слова, використано регулярні вирази. Наприклад, в скороченні «9/9п», що означає дев’ятий поверх дев’ятиповерхового будинку, літера «п» може часто зустрічатися в

тексті, через що з використанням словника можна замінити неправильні символи й отримати помилковий результат. Код з регулярним виразом, що вирішує цю проблему має такий вигляд:

```
match = re.search(r'\d+/\d+n', content)
    if match:
        content = content[:match.start()] + match.group().replace('n', ' поверх')
+ content[match.end():]
```

Також в деяких оголошеннях вказують ціну з використанням пробілу для розділення тисяч. Наприклад: «33 500$» Такий запис аналізатор сприймає як два окремих числа, тому використано регулярний вираз для того, щоб прибрати цей пробіл. Він має такий вигляд:

```
match = re.search(r'(\d)\s+(\d{3})\$', content)
if match:
  content =content[:match.start()]+match.group().replace(' ', '')
+content[match.end():]
```

Ще одна проблема виникає, коли в оголошенні є числові значення з комою ( або крапкою), тобто не цілі числа. Аналізатор розбиває цілу частину, кому й дробову частину на окремі токени, що призводить до некоректного створення залежностей між словами. Для вирішення цієї проблеми створено перевірку не лематизовнаго тексту на наявність чисел з комою. За допомогою регулярного виразу в тексті відбувається пошук не цілих чисел, потім вони записуються в масив, а в тексті замінюються на стрічку «0000000000000000000000000000000». Така кількість нулів зумовлена тим, що вона точно не траплятимуться в тексті, на відміну від «00» або «0000». Код з регулярним виразом має такий вигляд:

*myList = re.findall('\d+[.,]\d+', content)*

*content = re.sub('\d+[.,]\d+', ' 00000000000000000000000000000000 ', content)*

Після того, як аналізатор виконає роботу, числа знову вписуються на свої місця в тексті. Оскільки для аналізатора «00000000000000000000000000000000» – це один числовий токен, всі залежності в реченні створюються коректно і проведена заміна ніяк не впливає на результат.

## 4.2 Налаштування аналізатора

Для обробки української живої мови існує досить мало рішень, це пояснюється тим, що робота з NLP залежить від спеціального текстового корпусу.

Текстовий корпус – це електронне зібрання текстів природної мови, впорядковане, організоване й оформлене певним чином, призначене для наукового та практичного вивчення мови.[9] Найбільш вагомими є розмічені текстові корпуси, бо вони містять вручну прописану командою науковців морфологічну інформацію.

Серед наявних програмних рішень для обробки природної української мови найбільше переваг і можливостей присутні в моделі UDPipe, що працює на базі розміченого корпусу. У неї є свої недоліки, але з боку швидкості, зручності і якості роботи вона є найкращим інструментом, тому використовується саме цей засіб.[10]

UDPipe – це пайплайн для токенізації, теування, лематизації та розбору залежностей файлів CoNLL-U, який можна навчати. UDPipe має мовну діагностику й може навчатися на основі анотованих даних у форматі CoNLL-U. [11] CONLL-U –це стандартний формат, який використовується для анотування даних на рівні речення та на рівні слова/лексеми. Анотації у форматі CONLL-U відповідають таким вимогам:

1. Рядки слів містять анотації слова/токена в 10 полях, які містять інформацію про форму слова, лему, частину мови, універсальне відношення залежності ;
2. Порожні лінії позначають межі речень;
3. Рядки коментаря починаються з решітки (#).

UDPipe модель навчено на золотому стандарті. «Золотий морфосинтаксовий стандарт» – це текстовий корпус української мови, який розмічено повністю руками в декілька шарів: поділ на документи, абзаци, речення й токени; повна морфологія; синтакса залежностей. [12]

Для того, щоб отримати доступ до UDPipe для розробки парсера, використано готовий прикладний програмний інтерфейс (Application Programming Interface, API). Код, який надсилає запит до вебсервісу має такий вигляд:

```
files = {
        'data': content,
        'model': (None, 'ukrainian'),
        'tokenizer': (None, ''),
        'tagger': (None, ''),
        'parser': (None, ''),
      }
response=requests.post('http://lindat.mff.cuni.cz/services/udpipe/api/process',
files=files)
write =response.text.replace("\\n","\n").split("\n")
```

Запит має змінну «content» – це текст, який потрібно проаналізувати, а також вказано модель, яку потрібно використовувати, а саме: українську. За допомогою бібліотеки requests виконується запит із передачею параметрів файлів та

інформацією про тип обробки. У відповідь на запит, вебсервіс повертає результат обробки тексту, який зберігається в змінній write для подальшого використання.

### 4.3 Пошук адреси в оголошенні

Одним із важливих аспектів, що допомагає потенційним покупцям утвердити рішення щодо покупки квартири, є розташування об’єкта. Зазвичай в оголошеннях про нерухомість вказано місто, вулицю та подекуди номер будинку.

Для того, щоб у тексті оголошення знайти, в якому місті розташовано об’єкт нерухомості, створено масив з переліком міст України. Цей список взято зі сторінки Вікіпедії [13] і записано в змінну ukrainian_cities. Далі перевіряється чи наявні елементи масиву в лематизованому тексті. Якщо так, змінній findCity присвоюється значення елементу масиву, якщо ні, нічого не відбувається й місто не знайдено в тексті.

Пошук назви вулиці й номеру будинку реалізовано за допомогою розпізнавання іменованих сутностей, для цього використано бібліотеку spaCy, оскільки в UDPipe немає такого функціоналу.

spaCy — це безкоштовна бібліотека з відкритим кодом для розширеної обробки природної мови (NLP) на Python. spaCy може розпізнавати різні типи іменованих об'єктів у документі, звертаючись до моделі за прогнозом. Оскільки моделі є статистичними й сильно залежать від прикладів, на яких їх навчено, розпізнавання іменованих сутностей не завжди працює ідеально.[14]

Початковим кроком є завантаження моделі «uk_core_news_sm», ця модель містить лексичні, синтаксичні та семантичні компоненти, необхідні для обробки тексту українською мовою. Далі відбувається обробка тексту і за допомогою циклу for перевіряється чи містять іменовані сутності мітку «LOC», що позначає розташування. З оголошення, яке написано раніше, отримано такий результат:

*['вулиця  Кордуби', 'Житлова площа 17']*

Для того, щоб отримати назву вулиці, кожен елемент отриманого масиву перевіряється. Якщо знайдено слово «вулиця» чи «проспект» змінній, присвоюється значення без слова «вулиця» чи номера будинку.

Номер будинку може містити числа й літери, а також символи «/», наприклад: «вулиця Жилянська, 28», «вулиця Жилянська, 23Б», «вулиця Жилянська, 30/32».Для пошуку номера будинку використано регулярні визари, код має такий вигляд:

```
if(street.find('/')>0):
    streetNumber = "".join(re.findall('[0-9]+[а-яА-Я]*/[0-9]+[а-яА-Я]*',
street))
  else:
    streetNumber = "".join(re.findall('[0-9]+[а-яА-Я]*', street))
```

Також було помічено, що іноді аналізатор неправильно визначає іменовані сутності й адреса містить фразу «Гарне розташування», яку доволі часто пишуть після назви вулиці. Для вирішення цього використано функцію replace і фразу замінено на порожнє значення.

## 4.4 Пошук кімнатності в оголошенні

Кімнатність – це кількість кімнат у квартирі, ще один важливий аспект, який допомагає потенційним покупцям зрозуміти розмір житла.

Для того, щоб визначити, яка кімнатність квартири, спочатку потрібно знайти речення, яке містить стрічку «кімнат». Оскільки в реченні може бути написано «4 кімнатна» або «в квартирі 4 кімнати», обрано «кімнат».

Далі в знайденому реченні потрібно визначити, який номер має токен «кімнатний» чи «кімната» й перевірити чи є токен, що містить число і пов'язаний з

отриманим номером. Якщо такий токен є, змінній присвоюється знайдене значення. Код, який виконує потрібну перевірку має такий вигляд:

```
for i in roomsNumberSentence:
  if(roomsNumberSentence.get(i)[6]==wIDforRoomsNumber
      and len(re.findall('[0-9]+',roomsNumberSentence.get(i)[1]))>0):
    numberOfRooms="".join(re.findall('[0-9]+',roomsNumberSentence.get(i)[1]))
    break
```

Також, у деяких випадках, кімнатність квартири записують одним словом. Наприклад: «однокімнатна», «двокімнатна» чи «двохкімнатна». Для того, щоб знайти кімнатність у такому записі, потрібно перевірити чи є таке слово в тексті оголошення, якщо є, змінній присвоюється відповідне значення, тобто «однокімнатна» — 1, «двокімнатна», «двохкімнатна» — 2.

## 4.5 Пошук поверху в оголошенні

Для пошуку поверху в оголошенні спочатку потрібно знайти речення, що містить слово «поверх». Після цього в знайденому реченні відбувається проходження по всім токенам і ,якщо знайдено той, що пов'язано з отриманим номером і його частина мови визначена як «NUM» (число), змінній присвоюється отримане значення.

Також часто зустрічаються такі випадки написання поверху і поверховості будинку:

- Поверх: 9/9;
- Поверх: 9 з 9;
- Поверх: 9 із 9;
- 9 поверх з 9.

Тут відображено і поверх, і поверховість. Після того, як знайдено номер поверху, потрібно визначити чи наступний токен – це символ «/» або прийменники «з», «із». Якщо так, то потрібно перевірити наступний токен: його частина мови має бути визначена як «NUM». Якщо всі умови виконані змінній, яка позначає поверховість, потрібно присвоїти отриманий токен.

У випадку «9 поверх з 9», потрібно перевірити наступний токен після слова «поверх» замість числа й зробити ті ж перевірки, що зазначено в попередньому абзаці.

## 4.6 Пошук поверховості в оголошенні

Поверховість зазвичай в оголошеннях записують так: «4 поверховий будинок» або «4 поверхового будинку».

Тому, якщо на попередньому кроці не знайдено поверховість в оголошенні, спочатку потрібно знайти речення, що містить слово «поверховий». Після цього пройти по всіх токенах і ,якщо знайдено той, що пов'язано з отриманим номером і його частина мови визначено як «NUM», змінній присвоюється отримане значення.

## 4.7 Пошук площі в оголошенні

Площа – це одна з характеристик об’єкту нерухомості. Аби її знайти в оголошенні про нерухомість, спочатку потрібно знайти речення, яке містить слово «площа».

Оскільки в оголошеннях часто вказують не тільки загальну площу квартири, а ще площу кухні, потрібно знайти в реченні прикметник «загальний», «реальний» або «фактичний» і записати його номер у реченні. Після цього відбувається пошук токена «площа», який пов'язано з прикметником. Аналізатор зазвичай зв’язує токен «площа» з токеном, який позначає одиницю вимірювання, тобто «м2», тому далі

потрібно знайти номер цього токена. У кінці залишається знайти токен, який пов'язан з «м2» і частина мови якого визначено як «NUM».

Якщо не знайдено токен прикметника, потрібно одразу шукати токен «площа» і виконати подальші кроки так само.

Якщо виявиться, що токен «м2» не пов'язано з токеном «площа», потрібно знайти його номер одиниці вимірювання в реченні й далі шукати токен, що пов'язано з «м2» частина мови якого визначено як «NUM».

Також іноді в оголошеннях вказують площу так «66/42/13». Перше число означає загальну площу квартири. Для того, щоб її знайти використано регулярний вираз, код має такий вигляд:

```
match = re.search(r'\d+/\d+/\d+', content)
    if match:
        numbers = match.group(0).split("/")
        area = "".join(str(int(numbers[0])))
```

4.8 Пошук ціни в оголошенні

Ціна – це ще один важливий аспект, на який орієнтуються потенційні покупці при пошуку квартири. Зазвичай в оголошеннях ціну записують так:

- «35 тисяч доларів»;
- «35000$».

Для того, щоб знайти ціну, потрібно спочатку знайти речення, яке містить слова «тисяча доларів» або символ «$».

Далі потрібно перевірити речення на наявність токену «тисяча», якщо такий є, це перший варіант написання. Потрібно записати номер і знайти токен, який пов’язанj з отриманим номером і частина мови якого визначенj як «NUM». Наступним кроком потрібно перевірити чи отриманий токен менше тисячі q

домножити його на тисячу, щоб з «35 тисяч доларів» отримати «35000». Код, який це реалізує, має такий вигляд:

```
if(float(price)<1000):
        price = str(float(price)*1000)
```

Якщо токен «тисяча» не знайдено, потрібно перевірити речення на наявність токена «$». Далі потрібно записати номер токена в реченні й присвоїти змінній значення попереднього токена, який є числом.

## 4.9 Зберігання результату

Після обробки тексту одержані результати збережено у форматі JSON. JSON (JavaScript Object Notation) – це зручний формат обміну даними. Його легко аналізувати та генерувати як машинам, так і людям. JSON є повністю незалежним від мови програмування. Ці властивості роблять JSON ідеальною мовою обміну даними. [15] У мові програмування Python, для представлення результатів у форматі JSON, потрібно використати модуль «json». Кінцевий результат обробки тексту парсером має такий вигляд:

```
{'Місто': 'Львів',
'Номер будинку': '8',
'Вулиця': 'Рудненська',
'Кімнатність': '1',
'Поверх': '7',
'Поверховість': '9',
'Площа': '58.36',
'Ціна': '55000.0'}
```

## 4.10 Розробка GUI для демонстрації роботи парсера

Для наочної демонстрації роботи парсера розроблено графічний інтерфейс. Мова програмування Python має кілька бібліотек для розробки графічного інтерфейсу. В цій роботі використано Tkinter, оскільки він вбудований в стандартну бібліотеку Python, яка надає набір інструментів для створення графічних інтерфейсів.

Для початку потрібно імпортувати такі бібліотеки:

- tkinter для роботи з Tkinter;
- ttk для використання розширених елементів керування Tkinter, надає додаткові переваги, як-от: рендеринг шрифтів із згладжуванням у X11 і прозорість вікон;
- PIL бібліотека, яка призначена для роботи з зображеннями, надає широку підтримку форматів файлів  та можливості обробки зображень.

А також імпортувати власну функцію для обробки тексту оголошення. Для демонстрації роботи парсера вирішено створити вікно, яке містить три кнопки (кожна кнопка – це окреме оголошення, на якому демонструється робота парсера) і два текстових поля (у першому текстовому полі відображається оригінальний текст оголошення, а в другому – результат парсингу).

Зовнішній вигляд кнопок створено в «Figma» — векторному онлайн-сервісі розробки інтерфейсів та прототипування, збережено в форматі «PNG» і додано за допомогою бібліотеки «PIL», як зображення. Також для кожної кнопки написано функцію, яка зчитує текст файлу й заповнює текстові поля відповідною інформацією.

Графічний інтерфейс має такий вигляд:


Рисунок 4.1 – Початкове вікно графічного інтерфейсу

Після натиснення на кнопку, вікно має такий вигляд:

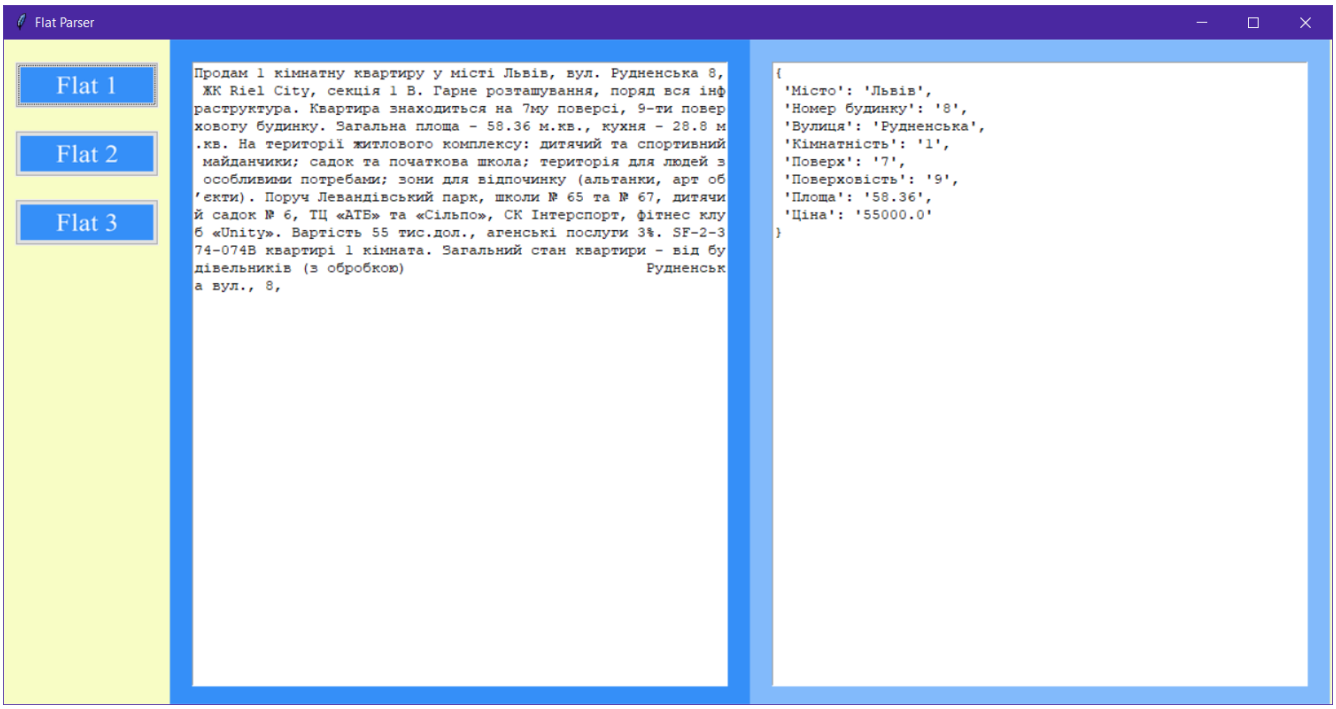


Рисунок 4.2 – Вікно парсера після натиснення на кнопку

## 4.11 Оцінка ефективності роботи парсера

Для того, щоб оцінити ефективність роботи парсера, проаналізовано 4284 оголошення й пораховано відношення кількості оголошень, у яких знайдено інформацію по кожному пункту до кількості оголошень, у яких ця інформація присутня. Отримано такі результати:

- знайдено місто в 99.92% оголошень;
- знайдено адресу в 76.12% оголошень;
- знайдено кімнатність в 85.57% оголошень;
- знайдено поверх в 88.58% оголошень;
- знайдено поверховість в 85.0% оголошень;
- знайдено площу в 100.0% оголошень;
- знайдено ціну в 85.20% оголошень.

# 5 Використання парсера

## 5.1 Модель для оцінки вартості нерухомості

У рамках розробки парсера для отримання даних про нерухомість, отримано значну кількість інформації про різні характеристики квартир, як-от:

- площа;
- кількість кімнат;
- розташування;
- поверх;
- поверховість;
- ціна.

Ці дані використано для створення моделі машинного навчання, яка здатна передбачати вартість квартири на основі її характеристик.

Машинне навчання (Machine Learning) – це підрозділ комп'ютерних наук, що забезпечує комп'ютери можливістю самостійно вчитися й виконувати певні дії без чітко вказаних програм. Як навчання, машина обробляє значні масиви вхідних даних і знаходить у них закономірності. Основними задачами машинного навчання є: розпізнавання, сортування, знаходження регресії. Кожна з цих задач знаходить своє застосування в різних сферах. [16]

Вибір моделі машинного навчання є важливим етапом у розробці проєкту, оскільки це визначає успіх і точність прогнозування. При виборі моделі потрібно враховувати наявні дані та їхні особливості, а також врахувати, яким чином модель може краще працювати з цими даними.

Однією з можливих моделей для задачі оцінки вартості квартир є лінійна регресія. Лінійна регресія є однією з найпоширеніших моделей машинного навчання. Її завдання полягає в тому, щоб знайти лінійну залежність між вхідними параметрами та вихідним значенням.

Іншою можливою моделлю є Decision Trees (дерева ухвалень рішень). Це метод машинного навчання, що використовується для вирішення задач класифікації та регресії. У Decision Trees дерево розбивається на вузли з відповідними умовами на вхідні параметри та класифікує дані за допомогою порівнянь.

Ще однією моделлю, яку можна розглянути для оцінки вартості квартир, є Random Forest. Ця модель машинного навчання комбінує декілька Decision Trees у єдину модель, а потім поєднує їхні результати для здійснення прогнозів. Основна перевага Random Forest полягає в здатності аналізувати великі набори даних із багатьма вхідними параметрами. Він дає змогу автоматично вибирати найбільш інформативні функції та уникати перенавчання. Крім того, Random Forest може продемонструвати значущі результати навіть без налаштування гіперпараметрів.

## 5.2 Аналіз даних для моделі

Оцінка вартості квартири є важливим завданням у сфері нерухомості. Вартість послуги з оцінки нерухомості залежить від оцінювача, є договірною та визначається в ході переговорів між опонентами. Також важливим фактором є місто. Так, ціни на оцінку нерухомості в Києві та Одесі найвищі – в межах 800-1500 грн, у Львові та Харкові трохи менше – до 1000 грн, а ось у Дніпрі послуги оцінювача обійдуться найдешевше – до 800 грн. Також існують спеціальні платні програми з оцінки нерухомості. [4]

Для того, щоб зрозуміти, які дані потрібні для оцінки вартості квартири, розглянуто онлайн калькулятор на сайті dom.ria. Це єдиний знайдений безкоштовний ресурс для оцінки вартості квартири. Розрахунок калькулятора здійснюється за алгоритмом ГІС «Увекон» на основі розташування й характеристик нерухомості та аналогічних об'єктів.

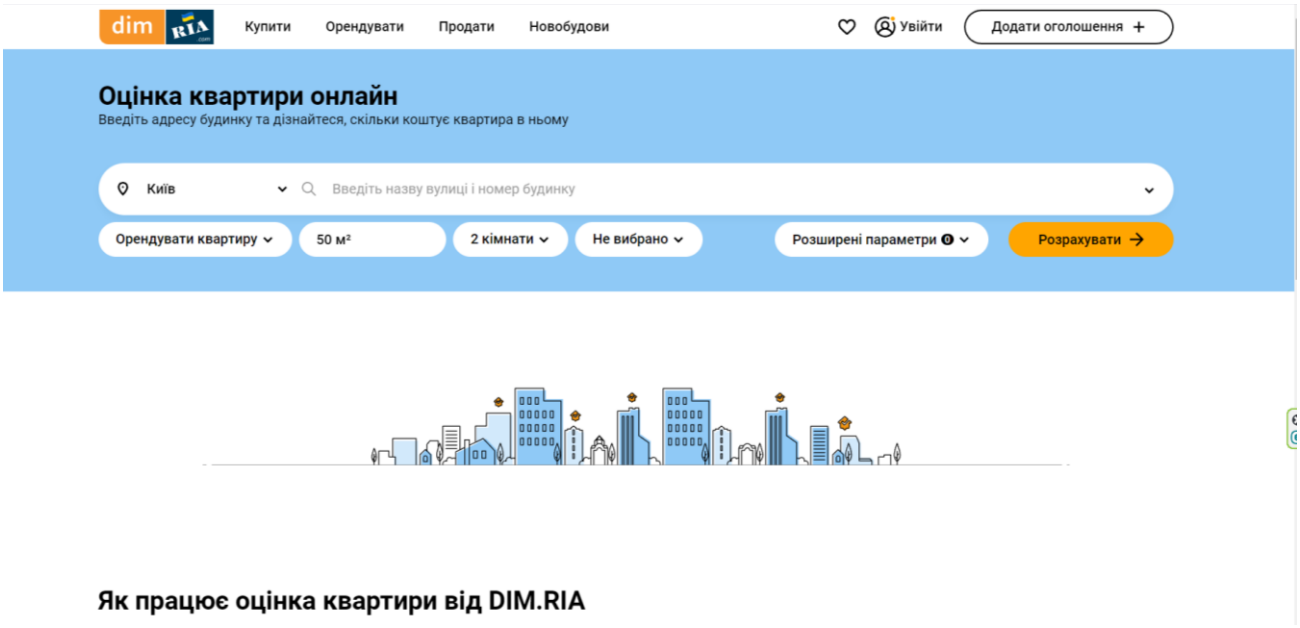


Рисунок 5.1 – Вебсторінка онлайн калькулятору dom.ria

Проаналізувавши цей вебресурс, можна побачити, що для оцінки вартості квартири достатньо характеристик, які отримано за допомогою парсера, а саме: місто, площа, кімнатність, поверх та поверховість.

## 5.3 Розробка моделі

Мова програмування Python зарекомендувала себе як одна з найпопулярніших мов для наукових досліджень. Завдяки своїй високорівневій інтерактивній структурі та розвиненій екосистемі наукових бібліотек вона є привабливим вибором для розробки алгоритмів та дослідницького аналізу даних.[17]

Scikit-learn – це модуль Python, що інтегрує широкий спектр найсучасніших алгоритмів машинного навчання для середньомасштабних контрольованих і неконтрольованих задач. Він зосереджений на тому, щоб зробити машинне навчання доступним для неспеціалістів, використовуючи універсальну мову

високого рівня. Акцент зроблено на простоті використання, продуктивності, документації та узгодженості API. [17]

Першим кроком розробки моделі є створення файлу з даними, які використано для навчання моделі. Для цього дані отримані з парсера зберігаються у файл «data.xlsx», вони мають такий вигляд:

| Місто | Адреса | Площа | К-сть кімнат | Поверх | Поверховість | Ціна |
|---|---|---|---|---|---|---|
| Львів | Вернадського Академіка | 83 | 3 | 1 | 11 | 129999 |
| Львів | Малоголосківська | 101 | 3 | 2 | 9 | 187400 |
| Львів | Коцюбинського | 94.9 | 3 | 1 | 3 | 385000 |
| Львів | Кондукторська | 60 | 1 | 5 | 5 | 96000 |

Таблиця 5.1 – Дані для навчання моделі

Далі імпортуємо бібліотеку «Pandas» для роботи з файлом. Pandas – це бібліотека з відкритим вихідним кодом, що надає високоефективні, прості у використанні структури даних та інструменти аналізу даних для мови програмування Python.[18] Наступним кроком створено DataFrame, в якому категоріальні змінні розширено за допомогою методу «get_dummies», тобто категоріальні змінні перетворено в числові, шляхом створення додаткових стовпців з прапорцями. Далі потрібно розділити дані: одній змінній присвоєно значення всіх стовпців, окрім «ціни», а другій змінній присвоєно значення стовпцю «ціна».

Для навчання моделі, отримані дані потрібно розділити на тренувальні та тестові, для цього використано функцію «train_test_split». Параметр test_size визначає відсоток даних, які призначено для тестування. У цьому випадку, 33% даних виділено для тестування, а решта 67% – для тренування моделі. Код має такий вигляд:

```
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)
```

Для використання Random Forest в бібліотеці scikit-learn, спочатку потрібно імпортувати відповідний клас. Основним класом, який використовується для роботи з Random Forest, є «RandomForestRegressor» для задач регресії. Код має такий вигляд:

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(X_train,y_train)
y_pred_train = model.predict(X_train)
y_pred = model.predict(X_test)
```

## 5.4 Оцінка моделі машинного навчання

Для того, щоб оцінити наскільки добре навчилась модель, пораховано максимальну й середню помилку між прогнозованими та фактичними значеннями. Також обчислюється коефіцієнт детермінації (R-squared) для моделі. Коефіцієнт детермінації – це метрика, яка використовується для вимірювання якості підлаштування моделі до даних.

Потрібно обчислити коефіцієнт детермінації для тренувальних даних. Цей показник вказує наскільки добре модель відповідає або "пояснює" залежність між вхідними змінними (X_train) й вихідними змінними (y_train). Чим ближче значення коефіцієнта детермінації до 1, тим краще модель підлаштовується до тренувальних даних.

Також потрібно обчислити коефіцієнт детермінації для тестових даних. Цей показник вказує, наскільки добре модель узагальнюється на нові дані, які вона раніше не бачила. Він вимірює, наскільки добре модель може передбачити вихідні значення (y_test) на основі вхідних значень (X_test), які не використовувалися під час навчання моделі.

Отримано такі результати:

- максимальна помилка – 209097$;
- середня помилка – 20236$;
- коефіцієнт детермінації для тренувальних даних – 88.09 %
- коефіцієнт детермінації для тестових даних – 33.51 %

На рисунку 5.2 зображено графік, який демонструє наскільки добре модель може передбачити вихідні значення. Чим ближче точка розташована до лінії, тим краще передбачена відповідь.

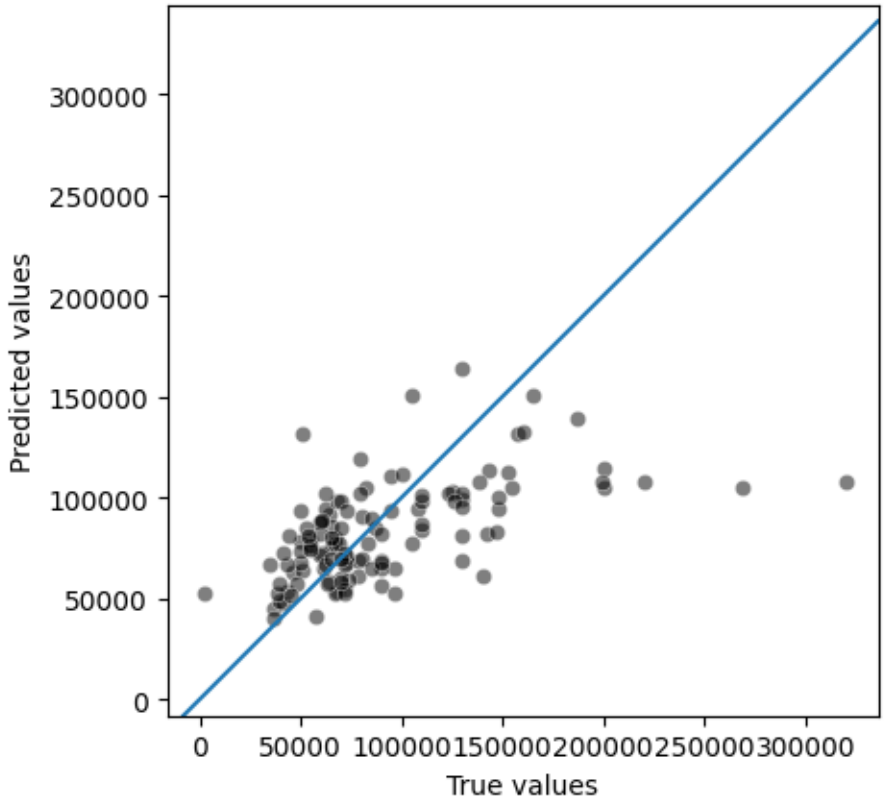


Рисунок 5.2 – Графік для порівняння передбачених відповідей і фактичних

Навчання моделі показало не вагомі результати, проте це можна виправити, навчаючи модель на більшій кількості даних. Складність навчання моделі полягає в тому, що тренувальні дані в найкращому випадку мають містити варіанти оголошень з усіма чинними адресами в місті й різною площею, кімнатністю.

# Висновки

Отже, в роботі проведено аналіз наявних на сьогодні можливостей з обробки природної української мови. Продемонстровано, що використання методів NLP дає змогу автоматизувати процес структурування текстів та отримувати значущі характеристики об'єктів нерухомості, а також вказано на наявні готові рішення, які використано для створення парсера.

Досліджено методи та стратегії для ефективного збору оголошень про нерухомість із вебресурсів.

Проведено частотний аналіз оголошень нерухомості з кількох сайтів, який демонструє головні характеристики квартир.

Також продемонстровано приклад використання парсера, а саме: створено модель для оцінки вартості нерухомості за допомогою машинного навчання.

Програмний застосунок дає змогу структурувати текст оголошень про нерухомість, виділяючи головні характеристики квартири, які потрібні потенційним покупцям для якісного пошуку квартири.

Перспективами розвитку парсера є розширення списку характеристик квартир, які зазначають в оголошеннях нерухомості, а також вдосконалення моделі, для створення ресурсу, який зможе якісно оцінювати вартість квартири.

Список літератури

1. Pak, Irina, and Phoey Lee Teh. "Text segmentation techniques: a critical review." Innovative Computing, Optimization and Its Applications: Modelling and Simulations (2018): 167-181.

2. Mitchell R. Web Scraping with Python: Collecting Data from the Modern Web. O'Reilly Media, Incorporated, 2015. 256 p.

3. Kaur, K., & Behal, S. Captcha and its techniques: a review. International Journal of Computer Science and Information Technologies. 2014. 5(5), 6341-6344.

4. DIM.RIA™ – вся нерухомість України. Продаж і оренда будь-якої нерухомості. DOM.RIA.com. URL: https://dom.ria.com/uk/

5. Нерухомість в Києві і Україні: продаж і оренда - RIELTOR.UA. Нерухомість в Києві і Україні: продаж і оренда - RIELTOR.UA. URL: https://rieltor.ua/

6. Дім Нерухомості Львів. Продаж і оренда у Львові квартир, приміщень, будинків | Дім Нерухомості Львів. URL: https://dim.lviv.ua/

7. Scrapy 2.9 documentation – Scrapy 2.9.0 documentation. Scrapy 2.9 documentation – Scrapy 2.9.0 documentation. URL: https://docs.scrapy.org/en/latest/

8. Intelligent Communication Technologies and Virtual Mobile Networks / ed. by G. Rajakumar et al. Singapore : Springer Nature Singapore, 2023. URL: https://doi.org/10.1007/978-981-19-1844-5

9. Демська, О. М. (2011). Текстовий корпус: ідея іншої форми. Київ: ВПЦ НаУКМА.

10. Смиш, О. (2020). Система для розв'язування задач з геометрії.

11. UDPipe. LINDAT/CLARIAH-CZ. URL: https://lindat.mff.cuni.cz/services/udpipe/

12. золотий стандарт. лабораторія української. URL: https://mova.institute/золотий_стандарт

13. Учасники проектів Вікімедіа. Міста України (за алфавітом) – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Міста_України_(за_алфавітом)

14. spaCy · Industrial-strength Natural Language Processing in Python. spaCy · Industrial-strength Natural Language Processing in Python. URL: https://spacy.io/

15. JSON. JSON. URL: https://www.json.org/json-en.html

16. Івченко, К. В. (2017). Машинне навчання та когнітивні обчислення. Матеріали міжнародної науково-практичної інтернет-конференції" Використання інноваційних технологій в процесі підготовки фахівців", Вінниця, 28-29 березня 2017 р.

17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

18. pandas documentation – pandas 2.0.1 documentation. pandas - Python Data Analysis Library. URL: https://pandas.pydata.org/docs/

Додаток А

(обов’язковий)

Перелік прийнятих скорочень

NLP (Natural Language Processing) – обробка природної мови

POS tagging (Part-of-speech tagging) – Розмічування частин мови

NER (Named entity recognition) – Розпізнавання іменованих сутностей