Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

“Інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання”

Виконав: студент 2 курсу, групи 2ІСТ-21м
спеціальності 126 – «Інформаційні системи та технології»
_____ Богачук А. Р.

Керівник: акад. НАПНУ, д.т.н., проф. каф. САІТ
_____ Мокін Б. І.
«02» _____ 2022 р.

Опонент: к.т.н., доц. каф. АІІТ
_____ Сторчак В. Г.
«16» _____ 2022 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«05» _____ 2022 р.

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – ІІ-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«16» 09 2022 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Богачуку Андрію Руслановичу

1. Тема роботи: "Інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання",
керівник роботи: Мокін Б. І., акад. НАПНУ, д.т.н., проф. каф. САІТ,
затверджені наказом закладу вищої освіти від «14» 09 2022 року №203
2. Строк подання студентом роботи «01» 12 2022 року
3. Вихідні дані до роботи:
Набір даних з платформи Kaggle, який містить інформацію по продажу будинків у Кінг-Каунті.
4. Зміст текстової частини:
   - аналіз предметної галузі передбачення ціни продажу будинків;
   - основні етапи виконання роботи та огляд вхідного набору даних;
   - побудова моделей та передбачення ціни продажу будинків;
   - економічна частина.
5. Перелік ілюстративного матеріалу:
   - блок-схема алгоритму інформаційної технології передбачення ціни на будинки;
   - теплова кореляційна матриця ознак;
   - перевірка на аномальні дані методами Matplotlib, Pandas та Seaborn;
   - інтерактивна мапа розташування будинків;
   - діаграма важливості ознак;
   - графік точності моделей.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
3	Мокін В. Б., д.т.н., проф. каф. САІТ	01.11.22р.	30.11.22р.
4	Бурєннікова Н. В., д.е.н., проф. каф. ЕПВМ	20.11.22р.	30.11.22р.

7. Дата видачі завдання «16» 09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі передбачення ціни продажу будинків	09.2022	
2	Основні етапи виконання роботи та огляд вхідного набору даних	10.2022	
3	Побудова моделей та передбачення ціни продажу будинків	11.2022	
4	Економічна частина	11.2022	
5	Оформлення матеріалів до захисту МКР	11.2022	

Студент _____ Богачук А. Р.

Керівник роботи _____ Мокін Б. І.

АНОТАЦІЯ

УДК 004.8:338+332.628

Богачук А. Р. Інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2022. 115 с.

Укр. мовою. Бібліогр.: 51 назв; рис.: 95; табл.: 5.

В магістерській кваліфікаційній роботі проведено аналіз предметної області передбачення ціни продажу будинків, попередньо запропоновано ознаки, які мають вплив на ціноутворення будинків. Здійснено огляд аналогічних рішень, запропоновано алгоритм створення ІТ передбачення ціни продажу будинків, на основі якого проведено розроблення ІТ. Виконано вибір та опис датасету, проведено попереднє очищення даних. Проведено розвідувальний аналіз даних, запропоновано правило фільтрації аномальних значень, обрано регресійні моделі машинного навчання, здійснено їх тренування та визначено оптимальну модель серед них, точність передбачення якої – 0.876, що є більшим за 0.82, як у найкращого аналога.

Ілюстративна частина складається з 6 плакатів.

У розділі економічної частини розглянуто питання про доцільність розроблення та впровадження інформаційної технології передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання.

Ключові слова: інформаційна технологія, розвідувальний аналіз даних, передбачення ціни, продаж будинків, ознаки, моделі машинного навчання.

ABSTRACT

Bohachuk A. R. Information technology of predicting the price of selling houses in King-County by methods of machine learning. Master's qualification work in specialty 126-information systems and technologies, educational and professional program-information technologies of data analysis and images. Vinnytsia: VNTU, 2022. 115 p.

In Ukrainian. language. Bibliogr: 51 titles; Fig.: 95; tab.: 7.

In the master's qualification work, an analysis of the subject area of predicting the sale price of houses was carried out, signs that have an impact on the pricing of houses are preliminarily proposed. A review of similar solutions was carried out, an algorithm for creating an IT prediction of the sale price of houses was proposed, on the basis of which the IT was developed. The selection and description of the dataset has been made, and preliminary data cleaning has been carried out. Exploratory data analysis was carried out, a filtering rule for anomalous values was proposed, regression models of machine learning were selected, their training was carried out and the optimal model among them was determined, the prediction accuracy of which is 0.876, which is more than 0.82, as the best analogue.

The illustrative part consists of 6 posters.

The section of the economic part deals with the question of the appropriateness of developing and implementing information technology for predicting the price of selling houses in the King-County by methods of machine learning.

Keywords: information technology, exploratory data analysis, price prediction, house, sales, features, machine learning models.

ЗМІСТ

ВСТУП

Актуальність теми. Завдання передбачення цін на нерухомість є дуже важливим сьогодні. У даний час можемо помітити багато будинків, які виставлено на продаж. Спостерігаються поодинокі випадки продажу будинків, які ще – у процесі будівництва, але більшість із них вже побудовані, у таких випадках ціна за квадратний метр будинку набирає суттєвої різниці. На ціноутворення будинку впливають багато факторів: місце розташування, район, площа, віддаленість від міста, рік побудови, кількість кімнат, кількість поверхів та багато інших факторів.

У зв’язку із цим виникає питання складання правильної ціни продажу будинку. Тема максимально точного передбачення ціни продажу будинків є актуальною у наш час, та напевне, не менш актуальною залишатиметься у найближчі роки, а можливо й у майбутньому.

Мета і завдання роботи. Метою даної роботи є підвищення точності передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання шляхом створення інформаційної технології передбачення цієї ціни.

Інформаційна технологія передбачення ціни продажу будинків складається з розв’язання таких задач:

– вибір оптимальних інформаційних технологій;

– вибір датасету, огляд основних ознак та попереднє очищення даних;

– проведення розвідувального аналізу даних;

– вибір оптимальної моделі, створення інформаційної технології та її застосування для передбачення даних.

Об’єктом дослідження магістерської кваліфікаційної роботи є процес передбачення ціни продажу будинків у Кінг-Каунті.

Предметом дослідження магістерської кваліфікаційної роботи є методи машинного навчання та інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті.

Методи дослідження. У дослідженні використовувались методи машинного навчання, регресійні моделі та моделі, які побудовані на основі дерев рішень, для передбачення ціни продажу будинків. У процесі виконання роботи проведено аналіз даних у системі Kaggle мовою програмування Python.

Новизна одержаних результатів. Наукова новизна полягає у тому, що дістала подальший розвиток інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті з використанням методів машинного навчання, яка дозволяє підвищити точність такого передбачення у порівнянні з аналогами.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи доповідались на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2022-2023 рр.).

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2022-2023 рр.) [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ПЕРЕДБАЧЕННЯ ЦІНИ ПРОДАЖУ БУДИНКІВ

1.1 Опис предметної області передбачення цін при продажі будинків

Продаж та купівля нерухомості, зокрема житла, будинків є надзвичайно важливими для нашого життя. Більшість людей звертаються в агентства нерухомості до ріелторів з метою придбання якісного житла та водночас по оптимально вигідній для покупця ціні.

Сьогодні можна спостерігати велику кількість будинків, які продаються. Деякі з них перебувають у процесі зведення, інші вже введені в експлуатацію. При цьому вартість квадратних метрів в об’єктах значно відрізняється. Ціни залежать від типу будівлі, міста, району, техніки будівництва, площі, планування, стану та багатьох інших факторів.

Оцінка будинку – послуга, без якої не обійтись у багатьох випадках. Фактично будь-які операції з нерухомим майном вимагають розрахунку його ринкової вартості [2].

Необхідність визначення того, скільки коштує будинок, потрібно в різних життєвих ситуаціях. Наприклад [2]:

– якщо оформляється спадщина, визначається реальна вартість будинку. Адже податкові зобов’язання за об’єкт, що успадковується, лягають на плечі спадкоємця. Занижена вартість також не схвалюється. У податківців можуть виникнути додаткові питання, що спричинить виплату додаткових податкових відрахувань;

– при внесенні власності до статутного капіталу підприємства;

– якщо будинку завдано шкоди;

– власник хоче застрахувати житло;

– оцінка вартості будинку під час розлучення;

– угоди купівлі-продажу тощо.

Оцінювачі нерухомості користуються двома підходами оцінювання.

Витратний. У цьому випадку ринкова вартість становитиме суму, яка дорівнює всім витратам, необхідним для будівництва будинку. Також слід звернути увагу, що фізичне зношування віднімають від загальної суми [2].

Порівняльний. У цьому підході оцінювач аналізує аналогічні пропозиції на ринку. Таким чином, встановлюється ринкова вартість об’єкта, який підлягає оцінці [2].

На ринкову вартість будинку впливає багато факторів, які слід враховувати [2].

Головні фактори оцінки житлового будинку [2]:

– місцеперебування (район, інфраструктура, транспортна розв’язка);
– загальна площа об’єкта;
– стан будинку, а також матеріали, з якого він виготовлений;
– рік будівництва;
– проведені інженерні комунікації (газ, вода, каналізація, світло, зв’язок).

Якщо ж йдеться про котеджне село або містечко, як, наприклад Кінг Каунті (округ штату Вашингтон, США), то на оцінку також впливатимуть внутрішня інфраструктура, віддаленість від центру, стан екології, наявність прибудинкової ділянки, охорони на території та навіть забудовник. Переваги за будь-якими пунктами підвищують цінність об’єкта, отже – і його вартість [2].

1.2 Розгляд технологій передбачення даних

Штучний інтелект – здатність інженерної системи обробляти, застосовувати та вдосконалювати здобуті знання та вміння [3].

Нейронні мережі розв’язують задачі, в яких треба проаналізувати велику кількість вхідних даних та багато ознак, зв'язки між якими невідомі. У традиційній статистиці методи обмежені строгими математичними термінами і є багато припущень щодо розподілу даних, на відміну від машинного навчання, де таких значних обмежень немає [4].

Також для навчання нейронної мережі потрібно ретельно підбирати вхідні дані, оскільки на цьому буде базуватися відповідь системи, для якісного підбору даних людині необхідний досвід і відповідна кваліфікація [4].

Машинне навчання – один з методів функціонування штучного інтелекту, а саме – практичної реалізації його можливостей шляхом створення алгоритмів для виявлення закономірностей під час аналізу великих даних, та їх подальше використання для самонавчання [5].

Мета машинного навчання – передбачити результат за вхідними даними. Чим різноманітніші вхідні дані, тим простіше машині знайти закономірності і тим точніший результат [6].

Кінцева мета машинного навчання – навчити модель вмінню правильно передбачати набір випадків, яких немає у навчальному наборі [7].

Метод, згідно з яким можна розрізняти різні категорії алгоритму – це тип результату, який очікується від певної системи навчання за допомогою машини.

Машинне навчання виконує такі завдання:

– визначення шахрайських дій – відстеження нетипових шаблонів у операціях з банківськими картками або рахунками;

– прогнозування – прогнозування майбутньої вартості акцій, валют, криптовалют, а також інших цінних паперів;

– розпізнавання зображень – визначення об'єктів та осіб (при необхідності, ідентифікація) у зображеннях та відео.

Для того, щоб навчити машину передбачати ціни на будинки перш за все необхідно мати:

– набір даних;

– ознаки;

– алгоритм.

Отже, для задачі передбачення ціни продажу будинків необхідно використовувати інформаційні технології, а саме методи машинного навчання.

1.3 Огляд та аналіз відомих аналогів

Існують вже готові системи пошуку та оцінювання вартості певного будинку чи квартири за вхідними ознаками, ці системи також відображають, як змінювалась ціна на будинок протягом обраного періоду часу, таким чином можна спостерігати за тим, як змінювалась ціна та виявити чи ціна є завищена, чи ні.

DOM.RIA – сайт, де продається і здається в оренду перевірена нерухомість по всій Україні. Оголошення з продажу новобудов від будівельних компаній та вторинної нерухомості від агентств та приватних осіб. Також присутня система оцінювання вартості житла на даному ресурсі за допомогою калькулятора вартості нерухомості [8].

Калькулятор вартості нерухомості – це сервіс від аналітичного центру DOM.RIA, за допомогою якого можна дізнатися приблизну ринкову вартість квартири за адресою. Для цього необхідно вказати лише її основні параметри, які істотно впливають на ціну житла. Просто вкажіть кількість кімнат і район, де розташована квартира [9].

Розрахунок вартості здійснюється за алгоритмом DOM.RIA на основі розташування і характеристик нерухомості та аналогічних об'єктів. Оцінка вартості є орієнтовною, оскільки ціна залежить від низки факторів, таких як стан житла, планування, вид з вікон тощо. Онлайн-оцінка допоможе визначити орієнтовну вартість усім, хто продає і купує нерухомість [10].

Калькулятор вартості нерухомості наведено на рисунку 1.1.

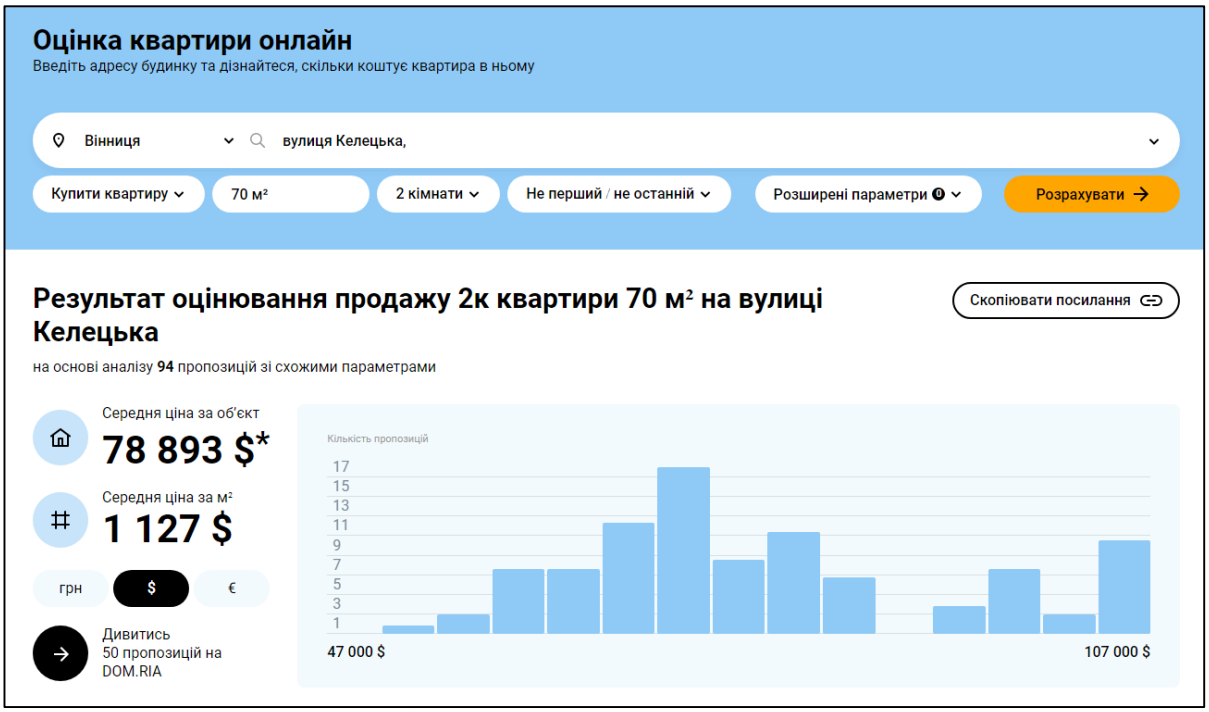

Рисунок 1.1 – Калькулятор вартості нерухомості від DOM.RIA

Особливостями даного аналога є те, що розрахунок орієнтовної ціни відбувається лише за наступними ознаками, такими як: місце розташування, площа, кількість кімнат та поверх. Для отримання більш точної ціни потрібно вказати більше ознак (параметрів) будинку, таких як: тип стін, ремонт, планування, техніка та меблі, тип будівлі (рис. 1.2).

Рисунок 1.2 – Ознаки для отримання більшої точності передбачення вартості

Варто зазначити, що онлайн калькулятор нерухомості підраховує середньо ринкову ціну виходячи з даних, які вказані в оголошеннях на DOM.RIA. Тому онлайн-оцінка квартири не є експертною, а спрямована на орієнтовний результат, оскільки на вартість житла може впливати цілий ряд факторів, таких як імідж і рейтинг району, стан ремонту, планування і багато іншого. До того ж, ціни можуть залежати від ситуації на ринку і від курсу валюти [9].

Як відомий аналог розглянемо та опишемо ноутбук у системі Kaggle під назвою «Predicting House Prices», автором якого є Burhan Y. Kiyakoglu [10].

У ноутбуці використано датасет, який містить 20 ознак та 21597 значень «House Sales in King County, USA». Цей набір даних містить ціни продажу будинків для Кінг -Каунті, що продаються в період з травня 2014 року по травень 2015 року [11].

У даній розробці для передбачення використовують різні моделі регресії та їх налаштування, як прості, так і складні, в результаті передбачення відбувається порівняння моделей за точністю.

У ноутбуці використано наступні основні бібліотеки мови Python: NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, Folium.

Після підключення бібліотек відбувається аналіз та візуалізація даних та їх попередня обробка, що дозволяє виявити аномальні дані чи викиди та відкинути непотрібні ознаки.

За допомогою бібліотеки Seaborn побудовано графіки «ящик з вусами», що наведено на рисунку 1.3.

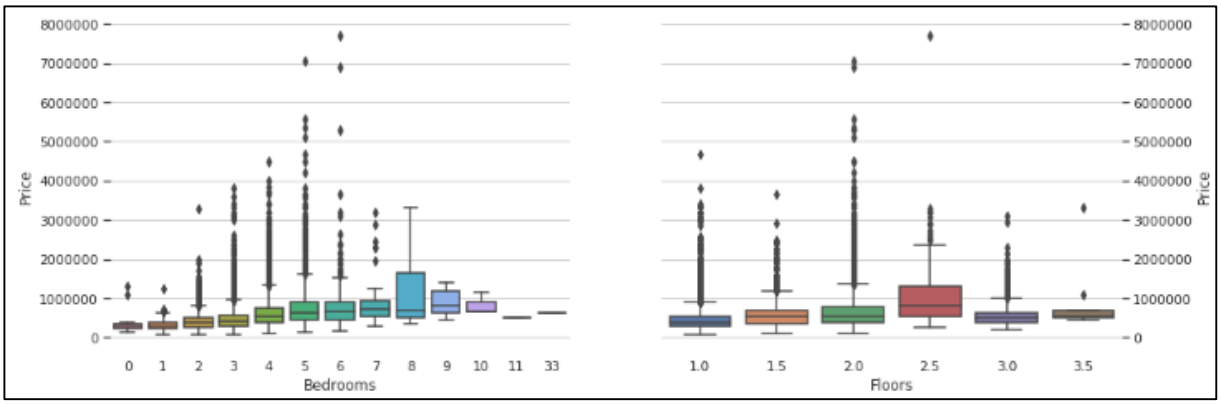

Рисунок 1.3 – Діаграма розкиду

З рисунка 1.3 видно, що є низка будинків, що мають певні особливості, ціна певних будинків набагато більша від ціни інших, або кількість спалень 33, тобто датасет містить викиди.

Автор аналога наводить кореляційну матрицю ознак за критерієм Пірсона (рис. 1.4).

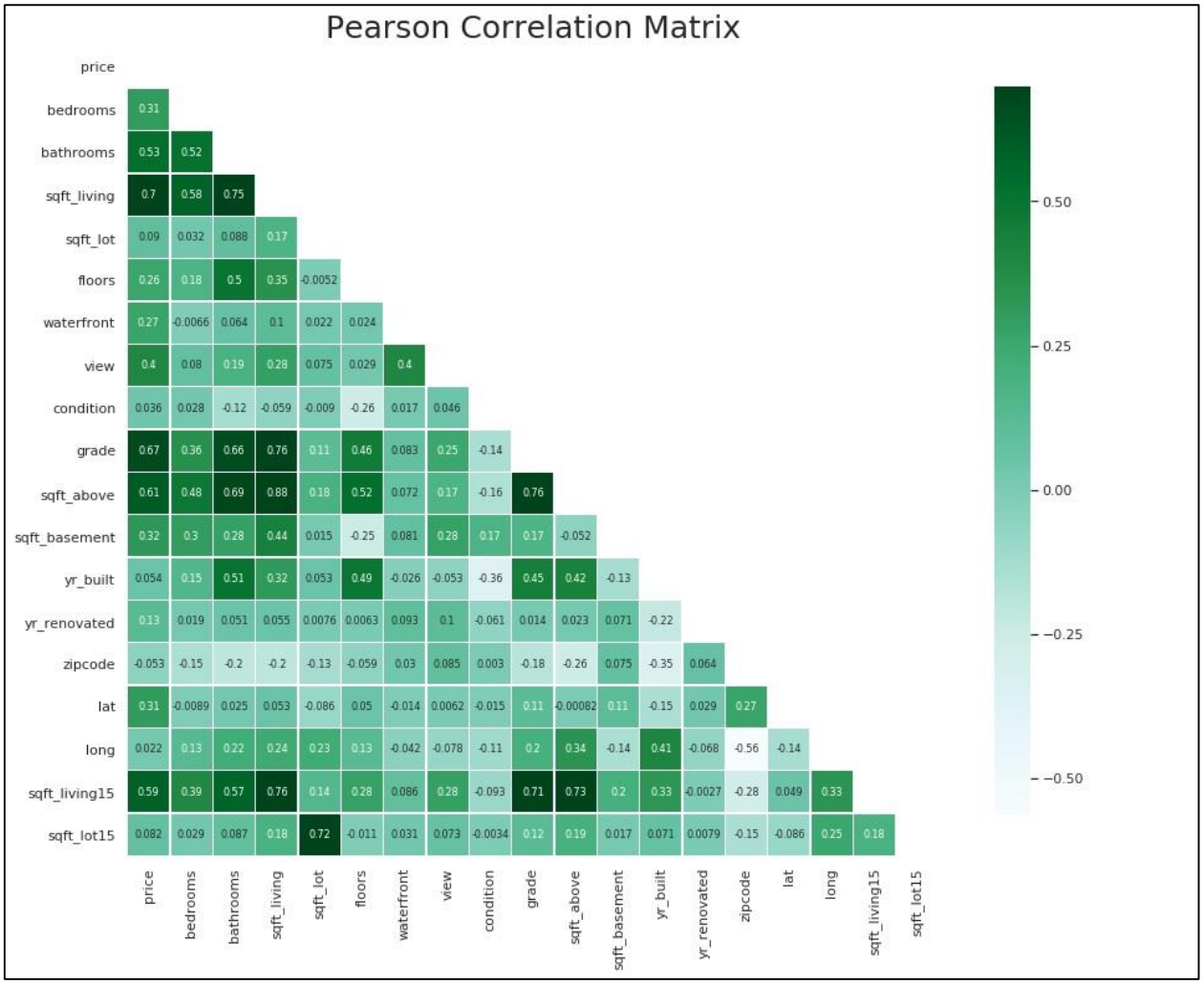

Рисунок 1.4 – Кореляційна матриця ознак

Ознаки sqt_above та sqt_living мають сильний зв’язок кореляцій, коефіцієнт кореляції між ними 0.88.

Наприклад, у випадку якщо спостерігається перенавчання моделі, можна видалити одну з ознак, оскільки вони дуже тісно корелюють між собою.

На етапі попередньої обробки даних для детального розуміння автор застосовує технологію групування даних до ознак «yr_built» та «yr_renovated».

Додано вік і час ремонту будинків, коли вони були продані. Крім того, автор розділив ці стовпці на інтервали. Візуалізацію згрупованих даних наведено на рисунках 1.5, 1.6.

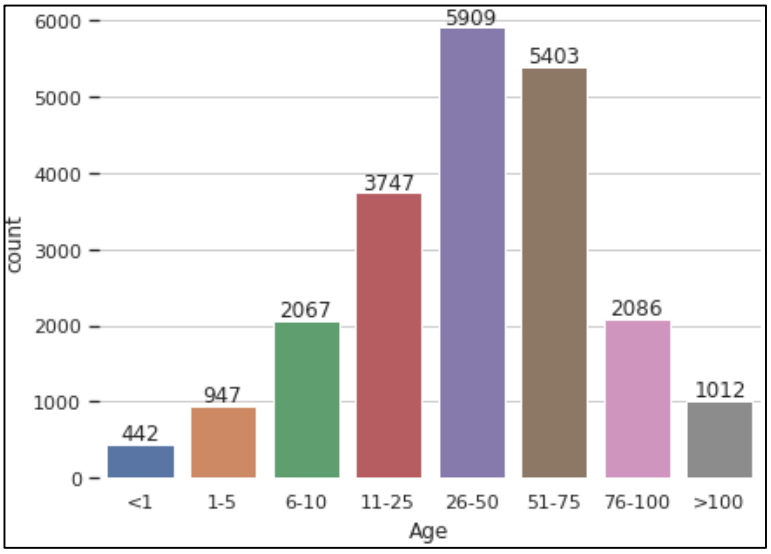

Рисунок 1.5 – Групування даних по ознаці «yr_built»

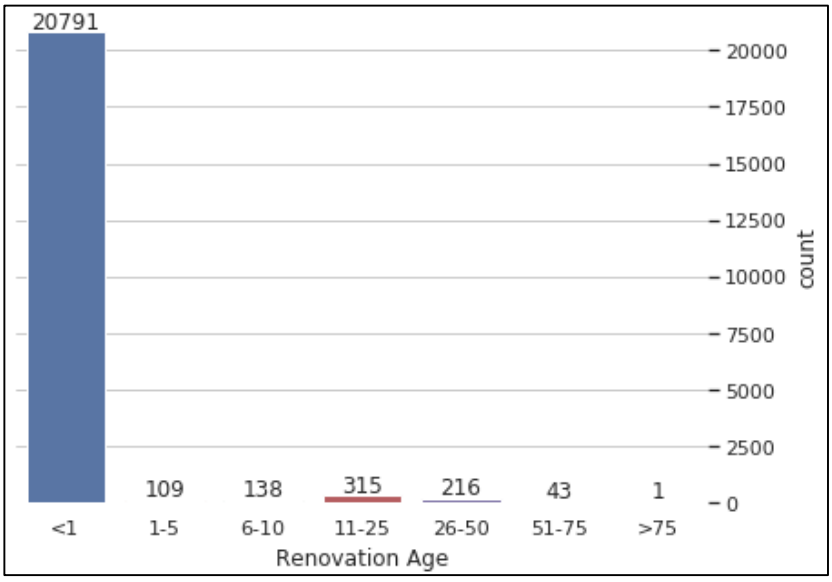

Рисунок 1.6 – Групування даних по ознаці «yr_renovated»

Таким чином, з діаграми на рисунку 1.5 видно 8 груп будинків по роках їх побудови. З діаграми, що на рисунку 1.6 бачимо 7 груп будинків по роках, коли відбувся останній ремонт будинку.

Автор аналога створив теплову мапу розташування будинків (рис. 1.7).

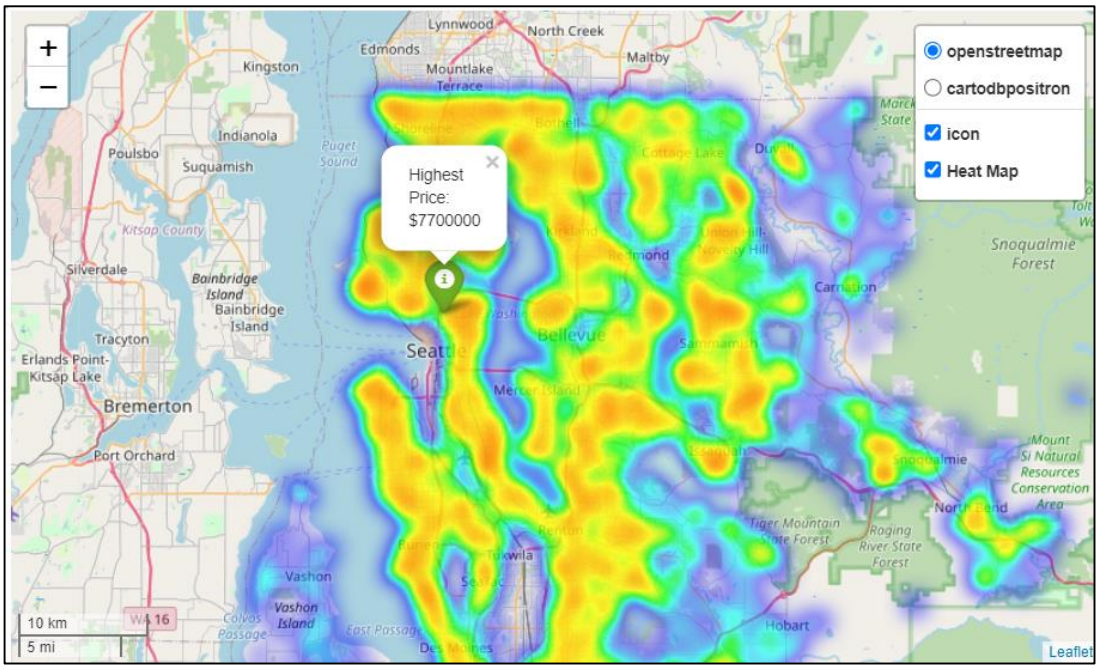

Рисунок 1.7 – Теплова мапа розташування будинків

Використовуючи стовпці широти та довготи, за допомогою бібліотеки “Folium” автор відобразив наведену теплову інтерактивну мапу, що на рисунку 1.7, яка є дуже корисною для людей, які погано знають Сіетл.

Крім того, автор аналога намалював тривимірний графік, щоб визначити зв’язок між видом, класом і роком будівництва (рис. 1.8).

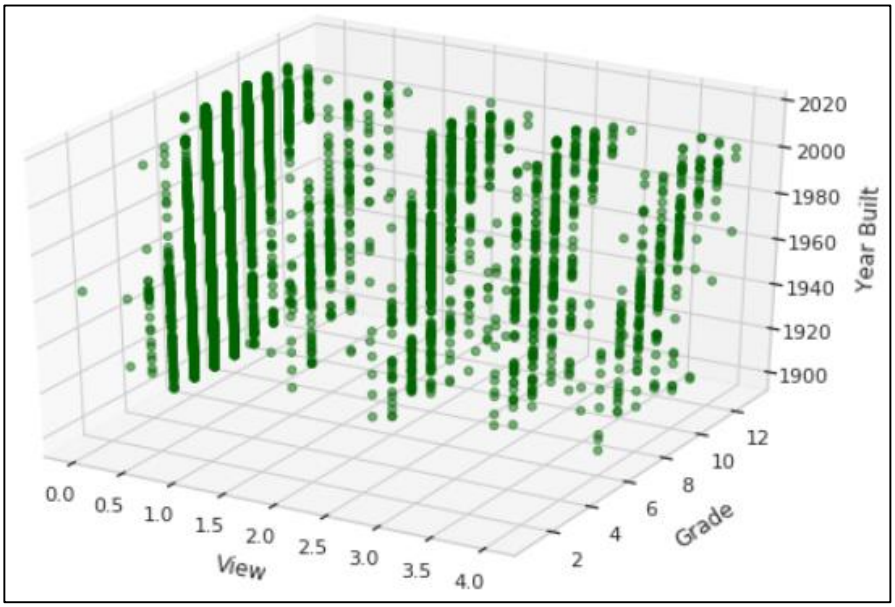

Рисунок 1.8 – Тривимірний графік (клас, оцінка, рік будівництва)

Наведена вище діаграма показує, що нові будинки мають кращі оцінки.

На етапі створення моделей в даному ноутбуці використовуються наступні моделі регресії: Ridge Regression, Lasso Regression, Polynomial Regression, k-NN Regression.

Таблиця оцінювання моделей та їх похибок наведена на рисунку 1.9.

	Model	Details	Root Mean Squared Error (RMSE)	R-squared (training)	Adjusted R-squared (training)	R-squared (test)	Adjusted R-squared (test)	5-Fold Cross Validation
0	Polynomial Regression	degree=2, all features, no preprocessing	151200.970	0.830	-	0.822	-	0.813
1	Polynomial Ridge Regression	alpha=50000, degree=2, all features	159872.572	0.810	-	0.801	-	0.791
2	Polynomial Lasso Regression	alpha=50000, degree=2, all features	166020.484	0.797	-	0.785	-	0.779
3	Polynomial Lasso Regression	alpha=1, degree=2, all features	166195.984	0.807	-	0.785	-	0.778
4	Polynomial Regression	degree=2, selected features, no preprocessing	190980.547	0.730	-	0.716	-	0.714
5	Multiple Regression-4	all features	191879.550	0.701	0.7	0.713	0.711	0.698
6	Ridge Regression	alpha=1, all features	191903.548	0.701	0.7	0.713	0.711	0.698
7	Lasso Regression	alpha=1, all features	191880.918	0.701	0.7	0.713	0.711	0.698
8	Lasso Regression	alpha=100, all features	192060.144	0.701	0.7	0.713	0.711	0.698
9	Multiple Regression-3	all features, no preprocessing	193693.989	0.698	0.697	0.708	0.707	0.695
10	Lasso Regression	alpha=1000, all features	193587.943	0.697	0.697	0.708	0.706	0.695

Рисунок 1.9 – Таблиця оцінювання моделей

На рисунку 1.9 наведено лише 10 найкращих моделей, звідси видно, що моделі, які використовують усі ознаки без попередньої обробки працюють точніше.

Отже, в результаті розгляду даного аналога виявлено, що на тесовому наборі даних модель поліноміальної регресії – найефективніша, оскільки її точність на тренувальному наборі даних – 0.83, на тестовому наборі даних точність передбачення – 0.81, але залежно від ситуації кожна з моделей може бути корисною.

Пропонуємо розглянути наступний аналог у системі Kaggle, ноутбук під назвою «House Price Predictions (R^2 0.82)» [12].

У ноутбуці використано датасет «House Sales in King County, USA» [11].

У цій розробці представлено та використано просту модель лінійної регресії в поєднанні з більш просунутими концепціями, такими як маніпулювання даними, розподіл даних, поліноміальні зв'язки тощо, а також показано способи розв'язку задачі передбачення ціни продажу будинків.

Для розуміння області передбачення автор наводить мапу, на якій наведено та виділено регіон Кінг-Каунті (1.10).

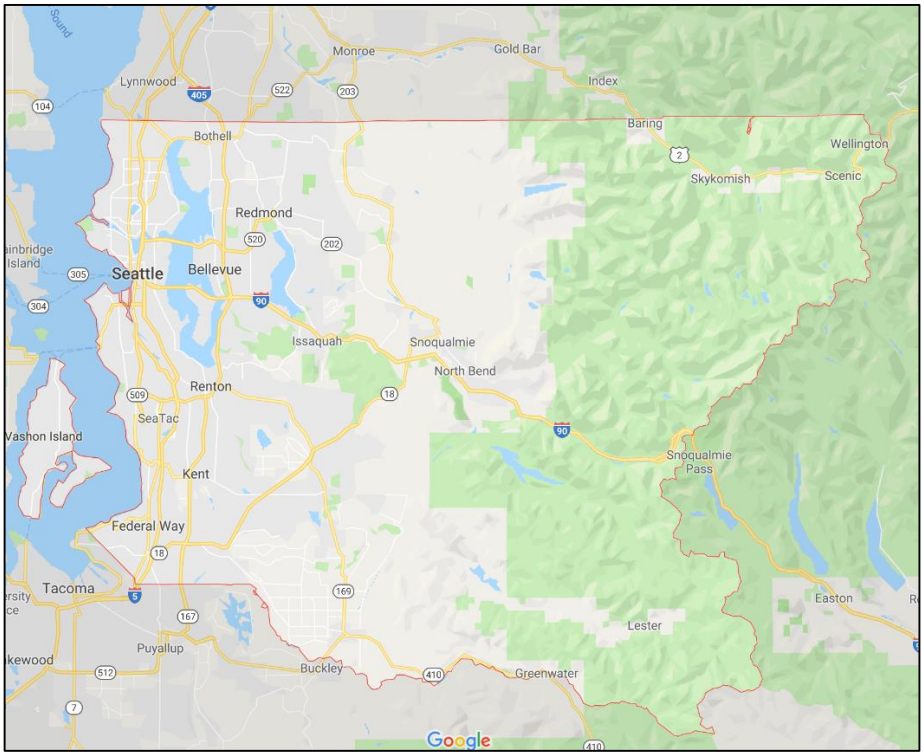

Рисунок 1.10 – Регіон Кінг-Каунті зображений на мапі

У роботі автор видаляє непотрібні ознаки (рис. 1.11).

```
df.drop(['id', 'date'], axis=1, inplace=True)
df.columns
```

```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'waterfront', 'view', 'condition', 'grade', 'sqft_above',
       'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long',
       'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

Рисунок 1.11 – Важливі ознаки набору даних

Стовпець ідентифікатора не дає жодного розуміння даних, як і дата, тому їх видалено, набір даних містить 19 характеристик будинку, а також 21613 спостережень.

У розділі розвідувального аналізу даних наведено інтерактивну мапу, на якій зображено ціну будинків відповідно до їх розташування (рис. 1.12).

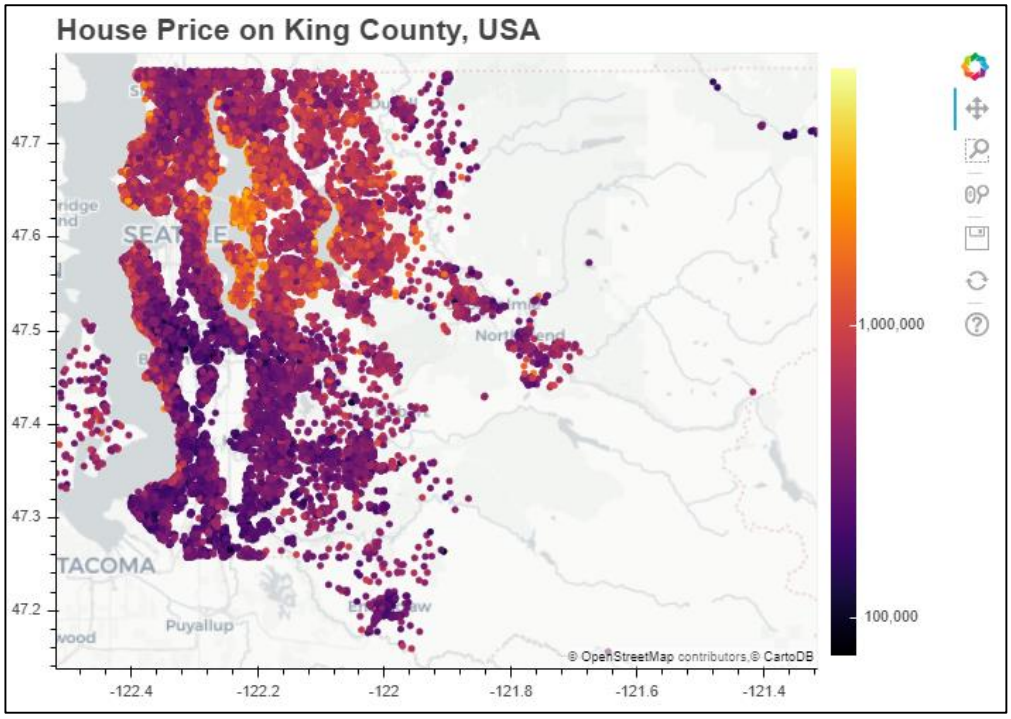

Рисунок 1.12 – Ціна будинків відповідно до розташування

З рисунка 1.12 видно, що північна частина регіону округу Кінг має більші ціни на житло. Чим ближче будинок до центру Сіетла, ціна на будинки зростає.

Виходячи із даного спостереження, автор ноутбука на етапі дослідження ознак створює нову ознаку, яка вимірює відстань в кілометрах від кожного будинку до центру Сіетла.

Фрагмент даних із новою ознакою “distance” наведено на рисунку 1.13

ade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15	distance
	1180	0	1955	0	98178	47.5112	-122.257	1340	5650	11.945427
	2170	400	1951	1991	98125	47.7210	-122.319	1690	7639	12.949082
	770	0	1933	0	98028	47.7379	-122.233	2720	8062	16.623098
	1050	910	1965	0	98136	47.5208	-122.393	1360	5000	10.348172
	1680	0	1987	0	98074	47.6168	-122.045	1800	7503	21.797975

Рисунок 1.13 – Фрагмент даних, що містить створену ознаку “distance”

Наступним етапом автор аналога виконує передбачення ціни на житло, використовуючи просту лінійну регресію та перехресну перевірку K-разів. Точність моделі лінійної регресії наведено на рисунку 1.14.

```
Using  k-fold cross-validation where k =  7 :
R2_adjusted of the test data, using a simple linear regression, is:  0.8219293314997594
```

Рисунок 1.14 – Результат передбачення, моделі лінійної регресії

Також за результатами передбачення наведено ознаки, які мають вплив на утворення ціни (рис. 1.15).

	coef
sqft	0.434615
distance	-0.415658
grade	0.269826
lat	0.242776
long	-0.113583
zipcode	-0.103421
view	0.089742
waterfront	0.082675
condition	0.080104
yr_built	-0.072453
bathrooms	0.071532
sqft_lot_comb	0.067986
bedrooms	-0.048334
yr_renovated	0.028770
sqft_basement	0.023327
floors	-0.002092

Рисунок 1.15 – Коефіцієнт впливу ознак на ціноутворення будинку

З рисунка 1.15 можна побачити коефіцієнти, які відсортовані за їх важливістю. Ознаки площі, відстані, класу та місця розташування мають найбільший вплив на ціну.

Негативні коефіцієнти означають, що збільшення цих ознак знижують ціну будинку. Наприклад, у міру зростання відстані будинків від Сіетла знижується, ціна на будинок знижується.

1.4 Висновки

У даному розділі виконано огляд важливості оцінювання будинків, охарактеризовано, що таке передбачення ціни будинків, розглянуто технології передбачення та технології машинного навчання.

Здійснено огляд відомих аналогів таких, як: DOM.RIA та ноутбуки у системі Kaggle з передбачення ціни на будинки. В результаті аналізу відомих аналогів можна підсумувати, що на ціноутворення будинків мають вплив наступні основні параметри, фактори та ознаки:

– місце розташування будинку (координати);
– площа будинку;
– кількість кімнат;
– кількість поверхів;
– рік побудови.

Результатом огляду та аналізу відомих аналогів у системі Kaggle визначено, що точність аналогічних рішень, які використовують подібні регресійні методи машинного навчання, складає 0.81 та 0.82.

2 ОСНОВНІ ЕТАПИ ВИКОНАННЯ РОБОТИ ТА ОГЛЯД ВХІДНОГО НАБОРУ ДАНИХ

2.1 Вибір оптимальних інформаційних технологій

На сьогоднішній день невід’ємною частиною технічної бази усіх організацій стали інформаційні системи та технології [13].

Інформація – це сукупність даних, які сприймають із навколишнього середовища (вхідна інформація), видають у навколишнє середовище (вихідна інформація) або зберігають всередині певної системи [14].

Інформаційна технологія – це система методів, процесів та способів використання обчислювальної техніки та систем зв'язку для створення, збору, передачі, пошуку, оброблення та поширення інформації з метою ефективної організації діяльності людей [15].

Як зазначено у розділі 1, для розв’язання задачі передбачення ціни продажу будинків доцільно використати інформаційну технологію машинного навчання.

Машинне навчання – це підгалузь штучного інтелекту в галузі інформатики, яка часто застосовує статистичні прийоми для надання комп'ютерам здатності «навчатися» з даних, без того, щоби бути програмованими явно [16].

Розв’язання будь-якої задачі аналізу та передбачення даних з використанням інтелектуальних методів машинного навчання зазвичай здійснюється у такі етапи [17]:

– очищення даних (виявлення і вилучення помилкових та аномальних даних);

– розвідувальний аналіз даних (EDA – Exploratory Data Analysis);

– видобування ознак (FE – Feature Engineering);

– ідентифікація та вибір оптимальної моделі передбачення даних.

Розглянувши аналогічні рішення, пропонується створення інформаційної технології передбачення ціни на будинки, яка буде використовувати алгоритм, етапи якого наведено у блок-схемі на рисунку 2.1.

Рисунок 2.1 – Блок-схема алгоритму інформаційної технології передбачення ціни на будинки

Кроки, які зазначено у блок-схемі алгоритму розв’язку задачі передбачення ціни на будинки, виконані та описані у наступних розділах роботи.

2.1.1 Вибір мови програмування та середовища розроблення

Найпоширенішою мовою для інтелектуального аналізу і передбачення даних в наш час є мова програмування Python [18].

Мова програмування Python найкраще підходить для роботи з методами машинного навчання, оскільки ця мова досить проста у порівнянні з іншими мовами. До того, Python має відмінні показники обробки даних [18].

Python – популярна мова програмування, створена в 1991 році Гвідо ван Россумом. Вона має ефективні структури даних високого рівня та простий, але ефективний підхід до об’єктно-орієнтованого програмування [19].

Python працює на декількох платформах (Windows, Mac, Linux, Raspberry Pi тощо) і має простий англійський синтаксис, що дозволяє розробникам писати програми з меншою кількістю рядків коду, ніж інші мови програмування. Python можна вважати процедурною, об'єктно-орієнтованою або функціональною мовою [18].

Однією з основних причин, чому Python використовується для машинного навчання, є те, що вона має багато фреймворків, які спрощують процес кодування та скорочують час розроблення.

Працюючи з Python, програмісту не потрібно приділяти великої уваги безпосередньому написанню коду: він може зосередити всю свою увагу на розв’язанні більш складних задач, пов’язаних з машинним навчанням. Простий синтаксис Python допомагає програмісту перевірити складні алгоритми з мінімальним часом реалізації [18].

Ще однією перевагою Python у машинному навчанні є його гнучкість, наприклад, програміст має вибір між об’єктно-орієнтованим підходом та сценаріями. Python допомагає поєднувати різні типи даних. Для того, щоб стати фахівцем в Machine learning, потрібно знати не тільки як систематизувати та аналізувати дані, як працюють базові алгоритми, вивчати реальні кейси, а й знати мову програмування Python, розібратися буде простіше [18].

Вищезазначені фактори пояснюють, чому Python так широко використовується у машинному навчанні, його простота та гнучкість допомагає працювати над складними алгоритмами з мінімальним часом реалізації [18].

Після визначення мови програмування необхідно обрати середовище, де буде відбуватись написання коду.

Середовище розроблення програмного забезпечення – це середовище, яке автоматизує або розширює підпрограми, що беруть участь у циклі розроблення програмного забезпечення. Це охоплює програмування в багатьох завданнях, таких як управління командою та проектами, а також великі завдання програмування, такі як управління конфігурацією [20].

На сьогоднішній день існує велика кількість середовищ для роботи з мовою програмування Python, а саме:

– PyCharm;
– Anaconda Jupiter Notebook;
– Kite;
– Spyder;
– IDLE;
– Visual Studio Code;
– Atom;
– PyDev, та багато інших.

Також існують онлайн середовища для написання коду мовою програмування Python, до таких середовищ можна віднести:

– Pythonanywhere;
– інші онлайн інтерпретатори Python;
– Kaggle Notebook.

У задачах машинного навчання доречно використати онлайн середовище розроблення від системи Kaggle, яке називається «Notebooks».

Система Kaggle дозволяє досліджувати та запускати код машинного навчання за допомогою Kaggle Notebooks, хмарного обчислювального середовища, яке забезпечує відтворюваний та спільний аналіз.

Варто зазначити, що використовуючи систему Kaggle Notebooks, можливо запустити код на хмарному обчислювальному середовищі із виділеними ресурсами оперативної пам’яті та до 20 ГБ вихідних даних із ноутбука можна зберегти на диск у /kaggle/working. Ці дані зберігаються автоматично і ви можете повторно використовувати їх.

Notebooks – це більше, ніж просто редактор коду. Це обчислювальне середовище, створене для полегшення відтворення наукової роботи з даними [21].

У IDE Notebooks у вас є доступ до інтерактивного сеансу, що працює в контейнері Docker із попередньо встановленими пакетами, можливість монтувати версійні джерела даних, настроюванні обчислювальні ресурси, такі як графічні процесори тощо [21].

Також є можливість спільного розроблення коду чи проекту, варто просто долучити учасника і надати статус співавтора та надати дозвіл на редагування коду у ноутбуці.

Перевагою Kaggle над іншими онлайн редакторами є те, що у системі є велика кількість наборів даних (датасетів) на різні тематики у вільному доступі, кожний, який бажає може працювати з цими даними, обговорювати, коментувати вже існуючі попередньо розроблені проекти (ноутбуки) та на основі вже наявних вдосконалити його, або за необхідності розробити власний ноутбук самотужки.

Отже, для виконання задачі по передбаченні ціни продажу будинків обрано мову розроблення та програмування Python, оскільки найбільше підходить для роботи з методами машинного навчання, у якості середовища розроблення програмного коду на мові Python обрано систему Kaggle IDE Notebooks, хмарних ресурсів для обчислення у безкоштовному режимі цілком достатньо, щоб виконати дану роботу.

2.1.2 Визначення виду машинного навчання

Класичне машинне навчання або Classical Machine Learning будується на класичних статистичних алгоритмах і розв’язує завдання, пов'язані з прийняттям рішень на основі даних [22].

Розрізняють наступні види машинного навчання [23]:

– контрольоване навчання (навчанням з учителем);
– неконтрольоване навчання (навчання без учителя);
– навчання з підкріпленням;
– нейронні мережі, глибоке навчання та інші.

Завданням машини при неконтрольованому навчанні (навчанні без учителя) є знайти зв'язок між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати класифікацію даних [23].

Навчання без учителя використовується, наприклад, в рекомендаційних системах, коли в інтернет-магазині на основі аналізу попередніх покупок покупцеві пропонуються товари, які можуть зацікавити його з більшою ймовірністю, ніж інші. До виду машинного навчання без учителя відноситься задача кластеризації [23].

Коротко кажучи, до виду машинного навчання без учителя відноситься задача, коли заздалегідь не відома цільова функція, а є лише різні вибірки даних для їх порівняння та аналізу.

Навчання з підкріпленням є окремим випадком контрольованого навчання, але вчителем в такому випадку є «середовище». Машина (її в цій ситуації часто називають «агент») не має попередньої інформації про середовище, але має можливість здійснювати в ній будь-які дії. Середовище реагує на ці дії і тим самим надає агенту дані, які дозволяють йому реагувати на них і вчитися. Фактично агент і середовище утворюють систему зі зворотним зв'язком [23].

Навчання з підкріпленням використовується, наприклад, в системах навігації для роботів, які навчаються уникати зіткнень з перешкодами шляхом

набуття досвіду, отримуючи зворотний зв'язок при кожному зіткненні також використовується в логістиці, при складанні графіків і плануванні завдань, при навчанні машини логічним іграм [23].

Класифікація і регресія – є завданнями, які виконуються за допомогою навчання з «учителем», також ще називають контрольованим навчанням.

Під терміном навчання з учителем розуміється те, що програміст, який навчає систему, розмічає дані, приводить машині певні приклади та спостерігає за її прогресом [22].

У ролі «вчителя» в даному випадку виступає людина, яка заздалегідь проставила «мітку», у нашому випадку «мітка» – це відома вірогідна ціна на будинки. Машина сама вибирає ознаки, за якими складає (передбачує) ціну на будинок. Тому надалі знайдений машиною алгоритм може бути швидко змінений на рішення іншої подібної за типом задачі [24].

Класифікація – найбільш популярна задача машинного навчання. Вона в чомусь схожа з тим, як дитина вчиться визначати форму і розмір предметів, складаючи їх у відокремлені купки [22].

Завдання класифікації є передбачення категорії об'єкта і поділ об'єктів згідно з визначеними та заданими наперед ознаками. Тобто, машина сортує дані за потрібними категоріями. Чудовим прикладом, де доцільно застосовувати класифікатор, є конкурс «Титанік» у системі Kaggle, коли маючи певній набір ознак потрібно передбачити бінарну ознаку, тобто відповісти на запитання вижив пасажир чи ні [22].

Але передбачення ціни не є бінарною ознакою, швидше за все це є передбаченням деякої цільової змінної, тому застосування класифікатора для даного типу завдання не є вірним підходом. Для розв’язання саме такого типу задач доречно використовувати регресію.

Регресія – це коли за заданим набором ознак необхідно передбачити якусь цільову змінну. Регресія запрограмована на роботу лише з числами [22].

Завданням регресії є передбачення місця на числовій прямій, наприклад: яка буде ціна на будинок або яким буде обсяг ринку певних товарів через 2 роки.

Отже, визначено, що задача передбачення ціни на будинки відноситься саме до виду машинного навчання з учителем (контрольоване навчання). А одним із найкращих варіантів розв’язання даної задачі є розв’язок за допомогою моделей регресії та моделей, які побудовані на основі дерев рішень.

2.1.3 Вибір та опис моделей машинного навчання

Існує багато моделей машинного навчання, але для реалізації даного завдання наведено та використано лише 7 популярних регресійних моделей машинного навчання, які не дуже вибагливі до ресурсів обчислювальної машини, але водночас з тим дають достатньо точне значення цільової величини, а саме:

– Random Forest Regressor;
– XGBRegressor;
– LightGBM;
– Bagging Regressor (Bootstrap Aggregation);
– Extra Trees Regressor;
– LinearRegression;
– MLPRegressor.

Random Forest (випадковий ліс) – це ансамблева техніка, здатна виконувати як регресію, а також класифікаційні завдання з використанням декількох дерев рішень і технікою, званою Bootstrap Aggregation, загальновідомою як пакування [25].

Основна ідея цього полягає в поєднанні декількох дерев рішень для визначення кінцевого результату, а не покладання на окремі дерева рішень [25].

До деяких мінусів даної моделі можна виділити те, що навчившись на одних і тих даних кілька разів – підхід передбачатиме трохи різні значення з різними можливостями. Тобто детермінованості, коли алгоритм видає завжди ті ж самі вихідні дані, ви не отримаєте.

Від цього не втечеш, тому що вибірка ансамблю вирішальних дерев відбувається випадковим чином. Для заморозки випадковості використовують ключове слово random_seed [25].

Алгоритм роботи моделі [25]:

– виберіть у навчальному наборі навмання K точок даних;

– побудуйте дерево рішень, пов'язане з цими точками даних K;

– оберіть кількість дерев, які ви хочете побудувати, і повторіть крок 1 і 2;

– для нової точки даних змусьте кожне з ваших дерев Ntree передбачити значення Y для точки даних та призначте новій точці даних середнє значення для всіх передбачених значень Y.

XGBoost розшифровується як "Extreme Gradient Boosting" – це реалізація машин для збільшення градієнта [26].

XGBoost – популярна контрольована модель машинного навчання з такими характеристиками, як швидкість обчислень, паралелізація та краща продуктивність [26].

Для роботи з регресією доцільно використовувати клас XGBRegressor пакета XGBoost.

LightGBM – це платформа для підвищення градієнта, що використовує алгоритм навчання на основі дерева.

Різниця моделі LightGBM від алгоритмів заснованих на деревах рішень полягає у тому, що LightGBM дерево росте вертикально, в той час, як інший алгоритм вирощує дерева в горизонтальному напрямку, тобто LightGBM виростає по дереву, в той час як інший алгоритм росте на рівні. Він вибере листя з максимальною втратою дельти, щоб рости. Під час вирощування одного й того ж листя алгоритм може зменшити більше втрат, ніж алгоритм, що обробляє рівень [27].

Нижче на діаграмах наведена реалізація LightGBM та інших прискорених алгоритмів (рис. 2.2, 2.3).

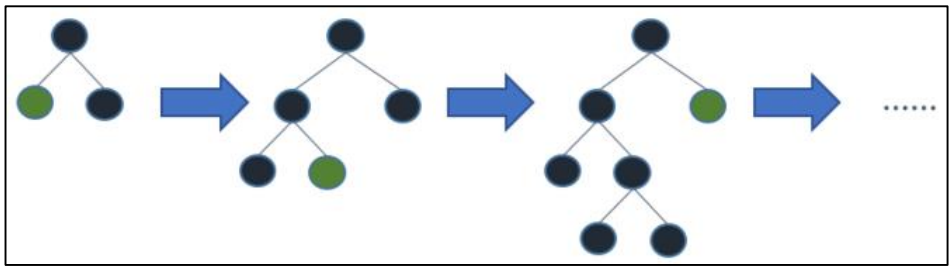

Рисунок 2.2 – Приклад роботи LightGBM

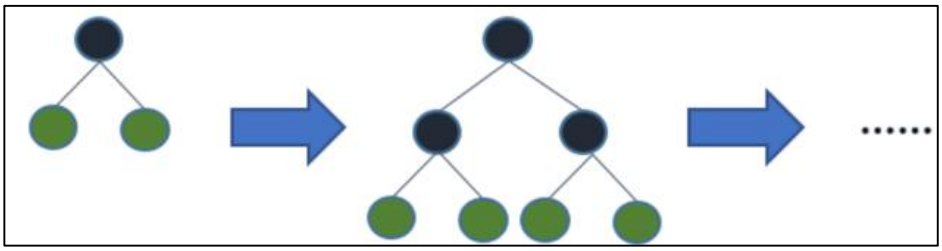

Рисунок 2.3 – Приклад роботи інших алгоритмів прискорення (бустингу)

Модель LightGBM варто використовувати на наборах даних, значень яких більше ніж 10 тисяч, тому що алгоритм дуже чутливий до перенавчання [27].

Реалізація LightGBM проста, єдині складнощі – це налаштування параметрів. LightGBM охоплює понад 100 параметрів, але для реалізації потрібно знати лише декілька основних параметрів [27].

1. Параметри управління [27]:

– max_depth (описує максимальну глибину дерева);

– min_data_in_leaf (це мінімальна кількість записів, які може мати «листя». Значення за замовчуванням – 20, воно є оптимальним);

– bagging_fraction (визначає частку даних, які будуть використовуватися для кожної ітерації, і, як правило, використовується для прискорення тренувань та уникнення перенавчання);

– early_stopping_round (цей параметр може допомогти прискорити аналіз. Модель припинить навчання, якщо один показник однієї валідації не поліпшиться в останніх ранніх етапах обходу. Це зменшить надмірні ітерації);

– lambda (лямбда вказує на регуляризацію, типове значення коливається від 0 до 1);

– min_gain_to_split (цей параметр описує мінімальний коефіцієнт посилення для розбиття, з його допомогою можна контролювати кількість корисних розщеплень на дереві);

– learn_rate (це визначає вплив кожного дерева на кінцевий результат, типові значення: 0.1, 0.001, 0.003…).

2. Параметри метрики [27]:

– MAE (середня абсолютна похибка);

– MSE (середня похибка квадрата);

– RMSE (корінне середньоквадратичне відхилення).

3. Параметри, які слід використовувати для підвищення ефективності моделі [27]:

– num_leaves (це основний параметр для контролю складності моделі дерева. В ідеалі значення num_leaves має бути меншим або рівним 2^(max_depth), значення, що перевищує це, призведе до перенавчання);

– min_data_in_leaf (якщо встановити велике значення, можна уникнути занадто глибокого росту дерева, але це може спричинити недонавчання);

– max_depth (використовують для явного обмеження глибини дерева).

4. Для підвищення кращої точності використовують наступні параметри [27]:

– max_bin (підвищують значення, але модель при цьому може працювати повільніше);

– learning_rate (використовують мале значення, але разом зі зменшенням цього параметру, збільшують параметр num_iterations).

Bagging Regressor (Bootstrap Aggregation) – це простий і дуже потужний метод ансамблю [28].

Під методом ансамблю розуміється поєднання прогнозів з декількох алгоритмів машинного навчання разом, щоб зробити більш точні прогнози, ніж будь-яка окрема модель [28].

Bootstrap Aggregation (агрегація завантажувальної програми) – це загальна процедура, яка може бути використана для зменшення дисперсії для тих алгоритмів, які мають велику дисперсію.

Алгоритм, який має велику дисперсію – це дерева рішень, як дерева класифікації та регресії [28].

Bagging – це застосування процедури Bootstrap до алгоритму машинного навчання з високою дисперсією, як правило, дерев рішень [28].

Як і самі дерева рішень, Baging може використовуватися для задач класифікації та регресії.

Extra Trees (Extremely Randomized Trees) – дуже схожий на Random Forest, оскільки він будує декілька дерев і розбиває вузли, використовуючи випадкові підмножини функцій, але з двома ключовими відмінностями: він не завантажує спостереження (тобто це вибірки без заміни), а вузли розбиваються на випадкові розбиття, не найкраще розпадається [29].

Отже, ExtraTrees [29]:

– створює кілька дерев з bootstrap = False за замовчуванням, що означає, що вибірки без заміни;

– вузли поділяються на основі випадкових розщеплень між випадковим набором функцій, вибраних у кожному вузлі.

LinearRegression – звичайна лінійна регресія методом найменших квадратів [30].

MLPRegressor – багатошаровий регресор персептрон. Ця модель оптимізує квадрат помилки за допомогою стохастичного градієнтного спуску [31].

Основні моменти багатошарового персептрона (MLP) у Scikit-learn [31]:

– на вихідному рівні немає функції активації;

– для сценаріїв регресії квадратична похибка є функцією втрат, а крос-ентропія є функцією втрат для класифікації;

– він може працювати як з одним, так і з кількома цільовими значеннями регресії;

– на відміну від інших популярних пакетів, таких як Keras, реалізація MLP у Scikit не підтримує GPU;

– неможливо точно налаштувати такі параметри, як різні функції активації, ініціалізатори ваги тощо для кожного шару.

2.1.4 Вибір та опис бібліотек Python

У мові програмування Python існує неймовірно велика кількість бібліотек. У даній роботі для її реалізації будуть використовуватись наступні бібліотеки:

Copy – бібліотека для копіювання, звичайне копіювання копіює лише посилання на об’єкт, натомість метод бібліотеки copy.deepcopy(x) повертає повну копію об’єкта у даному випадку x. Даний модуль надає загальні (поверхневі та глибинні) операції копіювання [32].

NumPy – це бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих (і дуже швидких) математичних функцій для операцій з цими масивами [33].

Pandas – це зручний і швидкий інструмент для роботи з даними, що володіє великим функціоналом [34].

Matplotlib – це бібліотека двовимірної графіки для мови програмування Python, за допомогою якої можна створювати високоякісні малюнки різних форматів [35].

Seaborn – бібліотека візуалізації даних Python, заснована на Matplotlib. Забезпечує інтерфейс високого рівня для малювання привабливої та інформативної статистичної графіки [36].

Plotly – бібліотека, яка дозволяє швидко візуалізувати дані та завдяки інтерактивності допомагає краще в них розібратися, тобто є можливість заглибитися в деталі при необхідності [37].

Mpl_toolkits – надає основні інструменти тривимірної побудови графіків (розсіювання, пошуку, лінії, сітки) [38].

SciPy – це пакет прикладних математичних процедур, заснований на розширенні Numpy Python. З SciPy інтерактивний сеанс Python перетворюється в таке ж повноцінне середовище обробки даних і прототип складних систем, як MATLAB, IDL, Octave, R-Lab і SciLab [39].

WordCloud – хмара, наповнена безліччю слів різного розміру, які показують частоту чи важливість кожного слова [40].

Folium – бібліотека для візуалізації географічних даних і інформації, яка містить координати та місця розташування [41].

Scikit-Learn – бібліотека для машинного навчання і прогнозної аналітики. Вона містить ряд методів, що охоплюють: алгоритми класифікації та регресії, кластеризації, валідацію і вибір моделей. Також її можна застосовувати для зменшення розмірності даних і виділення ознак [42].

Машинне навчання в Scikit-Learn полягає в тому, щоб імпортувати правильні модулі та запустити метод підбору моделі. Складніше очистити, відформатувати та підготувати дані, а також підібрати оптимальні вхідні значення і моделі. Тому перш ніж взятися за Scikit-Learn, потрібно, по-перше, відпрацювати навички роботи з Python і pandas, щоб навчитися якісно готувати дані, а по-друге, освоїти теорію і математичну основу різних моделей прогнозування та класифікації, щоб розуміти, що відбувається з даними при їх застосуванні [42].

Pandas_profiling – бібліотека для швидкого розвідувального аналізу даних. Результати її роботи виражаються не у вигляді якихось окремих показників, а в формі досить докладного HTML-звіту, що містить велику частину тих відомостей проаналізованих даних, які можливо буде потрібно знати перед тим, як приступати до більш щільної роботи з ними [43].

XGBoost (скорочення від EXtreme Gradient Boosting) – популярна бібліотека машинного навчання, що реалізує модель градієнтного бустингу, що представляє альтернативу регресійним методам і нейронних мереж. Метод полягає в створенні ансамблю послідовно уточнюючих один одного дерев рішень [44].

LightGBM – швидка, розподілена високопродуктивна платформа з градієнтним прискоренням (GBDT, GBRT, GBM) на основі алгоритмів дерева прийняття рішень. Бібліотека використовується для ранжирування, класифікації та багатьох інших завдань машинного навчання [45].

Warnings – бібліотека, яка видає попереджувальні повідомлення, як правило, видаються в ситуаціях, коли корисно попередити користувача про певний стан програми, коли ця умова (як правило) не вимагає збільшення винятку та припинення програми. Наприклад, можна створити попередження, коли програма використовує застарілий модуль [46].

2.2 Опис ознак вхідного набору даних

На етапі вибору датасету та огляду основних ознак пропонуємо використати дані США, Кінг-Каунті (по 21613 будинках) із датасету «House Sales in King County, USA» на базі платформи Kaggle, без обмежень на копіювання і використання [11].

Датасет містить наступні атрибути (ознаки) [11]:

– унікальний числовий номер, присвоєний кожному будинку, що продається (“id”);

– дата, коли будинок був розпроданий (“date”);

– ціна будинку у доларах (“price”);

– кількість спалень у будинку (“bedrooms”);

– кількість ванних кімнат (“bathrooms”);

– площа будинку у квадратних футах (“sqft_living ”);

– площа земельної ділянки (“sqft_lot”);

– кількість поверхів будинку (“floors”);

– чи є вид на набережну (“waterfront ”);

– чи переглядали будинок (“view”);

– стан будинку за шкалою від 1 до 5 (“condition”);

– загальна оцінка, на основі системи класифікації графства Кінг за шкалою від 1 до 11 (“grade”);

– площа будинку не враховуючи підвальне приміщення (“sqft_above”);

– площа підвального приміщення будинку (“sqft_basement”);

– рік побудови (“yr_built”);

– поштовий індекс будинку (“zipcode”);

– координати розташування будинку, широта та довгота (“lat”, “long”);

– площа житлового приміщення найближчих 15 сусідів (“sqft_living15”);

– площа земельних ділянок найближчих 15 сусідів (“sqft_lot15”).

Вигляд сторінки із датасетом у середовищі Kaggle наведено на рисунку 2.4.

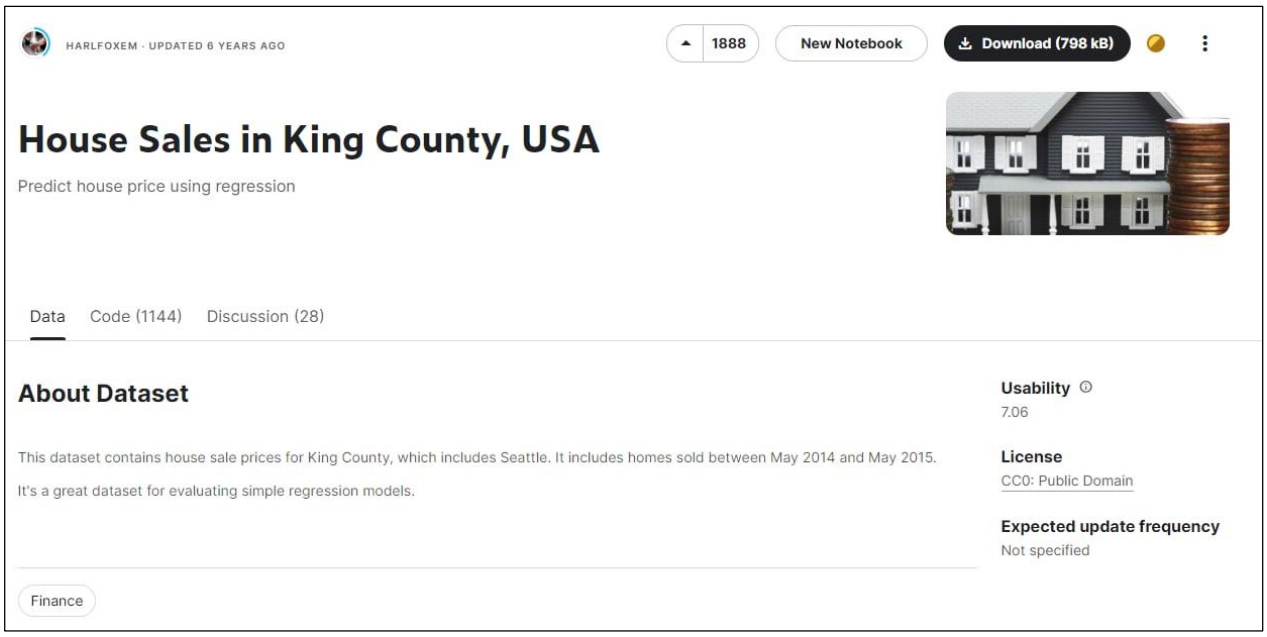

Рисунок 2.4 – Датасет у середовищі Kaggle

2.3 Зчитування набору даних

Першим етапом розв’язку будь-якої задачі машинного навчання є зчитування набору даних, але перед тим, як зчитати набір даних необхідно імпортувати необхідні бібліотеки, які будуть використовуватись на різних етапах розв’язання задачі.

Приклад імпортування бібліотек мови програмування Python в системі Kaggle наведено на рисунку 2.5.

```
import copy
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.style as style
import matplotlib.gridspec as gridspec
import seaborn as sns
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode,iplot
from mpl_toolkits import mplot3d
from scipy import stats
%matplotlib inline

# WordCloud
from wordcloud import WordCloud

# map visualization
import folium
from folium import plugins

# preprocessing
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV, Stratified
KFold
import pandas_profiling as pp

# models
from sklearn.linear_model import LinearRegression,LogisticRegression, SGDRegressor, RidgeCV
from sklearn.svm import SVR, LinearSVR
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.ensemble import BaggingRegressor
import sklearn.model_selection
from sklearn.model_selection import cross_val_predict as cvp
from sklearn import metrics
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder, StandardScaler
from scipy.stats import pearsonr

import xgboost as xgb
import lightgbm as lgb

# model tuning
from hyperopt import STATUS_OK, Trials, fmin, hp, tpe, space_eval

import warnings
warnings.filterwarnings("ignore")
```

Рисунок 2.5 – Імпорт бібліотек, які будуть використовуватись на різних етапах розв’язання задачі

Зчитування вхідного набору даних відбувається за допомогою бібліотеки pandas та її методу read_csv. Приклад зчитування даних та їх відображення наведено на рисунках 2.6, 2.7.

```
features = ['price','bedrooms','bathrooms','sqft_living','sqft_lot','floors','waterfront','vie
w',
            'condition','grade','sqft_above','sqft_basement','yr_built','yr_renovated',
            'zipcode','lat','long','sqft_living15','sqft_lot15']

train0 = pd.read_csv('../input/housesalesprediction/kc_house_data.csv')
train0 = train0[features]
train0.head(5)
```

Рисунок 2.6 – Фрагмент коду для зчитування та відображення набору даних з наперед вказаними ознаками

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_b
0	221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0
1	538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	400
2	180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	0
3	604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	910
4	510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	0

Рисунок 2.7 – Фрагмент зчитаних даних

Наступним етапом виводимо загальну інформацію по кожній ознаці набору даних, за допомогою команди info(). Загальну інформацію зчитаного набору даних наведено на рисунку 2.8.

```
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 19 columns):
price            21613 non-null float64
bedrooms         21613 non-null int64
bathrooms        21613 non-null float64
sqft_living      21613 non-null int64
sqft_lot         21613 non-null int64
floors           21613 non-null float64
waterfront       21613 non-null int64
view             21613 non-null int64
condition        21613 non-null int64
grade            21613 non-null int64
sqft_above       21613 non-null int64
sqft_basement    21613 non-null int64
yr_built         21613 non-null int64
yr_renovated     21613 non-null int64
zipcode          21613 non-null int64
lat              21613 non-null float64
long             21613 non-null float64
sqft_living15    21613 non-null int64
sqft_lot15       21613 non-null int64
dtypes: float64(5), int64(14)
memory usage: 3.1 MB
```

Рисунок 2.8 – Інформація про зчитаний набір даних

З рисунка 2.8 видно, що наш набір даних має 21612 значень, усі ознаки мають числовий тип значень, деякі ознаки мають тип значення з плаваючою крапкою. Пустих значень не спостерігається.

Приведемо дані, які мають дробову частину, до цілочислового типу даних (рис. 2.9).

```
train0['price'] = (train0['price']).astype(int)
train0['floors'] = (train0['floors']).astype(int)
train0['bedrooms'] = (train0['bedrooms']).astype(int)
```

Рисунок 2.9 – Приведення даних до цілочислового типу

Отже, на даному етапі набір даних задовольняє усі вимоги.

2.4 Фільтрація викидів та аномальних даних

Для елементарного фільтрування аномальних даних використано функцію describe(), дана функція виводить елементарні одиниці статистки, особливістю даної функції є те, що ми можемо подивитись на дані, взявши певний відсоток від них, ці відсотки також ще називають квантилями, якщо відсоток складає чверть, тобто 25% чи 75% його прийнято називати квартилем.

Першим етапом подивимось, як корелюють ознаки з нашою цільовою ознакою (ціною). Для цього виведемо датафрейм, у якому буде безпосередньо наведено значення оцінки кореляції кожної ознаки з ціною.

Створення та вигляд датафрейму наведено на рисунку 2.10 – 2.12.

```
featuress = train0.iloc[:,3:].columns.tolist()
target = train0.iloc[:,0].name
correlations = {}
for f in featuress:
    data_temp = train0[[f,target]]
    x1 = data_temp[f].values
    x2 = data_temp[target].values
    key = f + ' vs ' + target
    correlations[key] = pearsonr(x1,x2)[0]
```

Рисунок 2.10 – Створення стовпця, який містить назви характеристик

```
data_correlations = pd.DataFrame(correlations, index=['Value']).T
data_correlations.loc[data_correlations['Value'].abs().sort_values(ascending=False).index]
```

Рисунок 2.11 – Виведення усіх співвідносних значень функції щодо ціни, яка є цільовою змінною

	Value
sqft_living vs price	0.702035
grade vs price	0.667434
sqft_above vs price	0.605567
sqft_living15 vs price	0.585379
view vs price	0.397293
sqft_basement vs price	0.323816
lat vs price	0.307003
waterfront vs price	0.266369
floors vs price	0.237211
yr_renovated vs price	0.126434
sqft_lot vs price	0.089661
sqft_lot15 vs price	0.082447
yr_built vs price	0.054012
zipcode vs price	-0.053203
condition vs price	0.036362
long vs price	0.021626

Рисунок 2.12 – Кореляція значень функції з ціною продажу

З рисунка 2.12 видно, що значення кореляції між поштовим індексом та ціною продажу співвідноситься негативно, тому ми можемо відкинути його для передбачення ціни продажу.

Виведення елементарної статистики з наперед заданими квантилями наведено на рисунку 2.13.

```
train0.describe(percentiles=[.01, .05, .1, .5, .9, .92, .93, .94, .96, .97, .99])
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition
count	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000
mean	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.446213	3.409430
std	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.551894	0.650743
min	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	1.000000
1%	1.535004e+05	2.000000	1.000000	720.000000	1.013120e+03	1.000000	3.000000
5%	2.100000e+05	2.000000	1.000000	940.000000	1.800000e+03	1.000000	3.000000
10%	2.450000e+05	2.000000	1.000000	1090.000000	3.322200e+03	1.000000	3.000000
50%	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.000000	3.000000
90%	8.870000e+05	4.000000	3.000000	3250.000000	2.139760e+04	2.000000	4.000000
92%	9.500000e+05	5.000000	3.250000	3420.000000	2.851660e+04	2.000000	4.000000
93%	9.980000e+05	5.000000	3.250000	3510.000000	3.484832e+04	2.000000	5.000000
94%	1.063560e+06	5.000000	3.250000	3630.000000	3.768116e+04	2.000000	5.000000
96%	1.259040e+06	5.000000	3.500000	3920.000000	5.065816e+04	2.000000	5.000000
97%	1.388000e+06	5.000000	3.500000	4140.000000	6.743684e+04	2.000000	5.000000
99%	1.964400e+06	6.000000	4.250000	4978.800000	2.130080e+05	3.000000	5.000000
max	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.000000	5.000000

Рисунок 2.13 – Фрагмент елементарних статистичні одиниць

З рисунка 2.13 видно досить великий перепад між мінімальною ціною, та одним відсотком від неї, тому доцільно буде відфільтрувати аномальний «шматок» значень, аномальні дані негативно впливають на результат навчання моделі та на саме передбачення.

Усі інші виявленні аномальні дані відфільтровано по певному «порогу» наведено на рисунку 2.14.

```
train0 = train0[(
                (train0['price'] <= 1000000) &
                (train0['price'] > 170000) &
                (train0['bathrooms'] <= 4) &
                (train0['condition'] > 2.5) &
                (train0['grade'] != 4) &
                (train0['sqft_lot15'] > 1300) &
                (train0['sqft_lot15'] < 44000) &
                (train0['sqft_lot'] > 1500) &
                (train0['sqft_lot'] < 70000) &
                (train0['sqft_living'] > 700) &
                (train0['yr_built'] > 1925) &
                (train0['bedrooms'] > 0) &
                (train0['bedrooms'] < 7)
                )]
```

Рисунок 2.14 – Фільтрування аномальних даних

Також виявлено, що будинків, стан яких, по 5 бальній шкалі менший за 2.5 не так багато, серед 21411 всього 202 будинки, тому прийнято рішення відокремити їх, тобто не враховувати на наступних етапах (рис. 2.15).

```
(train0['condition'] > 2.5).value_counts()

True     21411
False      202
Name: condition, dtype: int64
```

Рисунок 2.15 – Обмеження за станом будинку

Кількість будинків, які мають загальну оцінку 4 по 11 бальній системі оцінювання всього 29, тому не враховуємо їх на наступних етапах (рис. 2.16).

```
(train0['grade'] == 4).value_counts()

False    21584
True        29
Name: grade, dtype: int64
```

Рисунок 2.16 – Обмеження за оцінкою

Також виявлено ряд ознак, які мають велику кількість нульових значень, або не несуть ніякої цінності при передбаченні, тому прийнято рішенні вилучити їх з набору даних (рис. 2.17).

```
drop_columns = ['zipcode', 'view', 'waterfront', 'yr_renovated']
train0 = train0.drop(columns = drop_columns)
```

Рисунок 2.17 – Вилучення ряду ознак з набору даних

Після застосування фільтрації та відокремлення аномальних частин ознак, набір даних містить 15825 значень, тобто вилучено 5784 стрічок, які містили аномальні дані.

2.5 Розвідувальний аналіз даних

Розвідувальний аналіз даних – це системний аналіз основних властивостей даних, знаходження в них загальних закономірностей, розподілів і аномалій, побудова початкових моделей [47].

Швидкий і досить потужний EDA на Python здійснює бібліотека pandas_profiling [47].

Побудуємо кореляційну матрицю за критерієм Пірсона (рис. 2.18, 2.19).

```
str_list = [] # empty list to contain columns with strings (words)
for colname, colvalue in train0.iteritems():
    if type(colvalue[1]) == str:
         str_list.append(colname)
# Get to the numeric columns by inversion
num_list = train0.columns.difference(str_list)
# Create Dataframe containing only numerical features
house_num = train0[num_list]
f, ax = plt.subplots(figsize=(16, 12))
plt.title('Pearson Correlation of features')
# Draw the heatmap using seaborn
sns.heatmap(house_num.astype(float).corr(),linewidths=0.25,vmax=1.0, square=True, cmap="cubehel
ix", linecolor='k', annot=True)
```

Рисунок 2.18 – Фрагмент коду для побудови кореляційної матриці

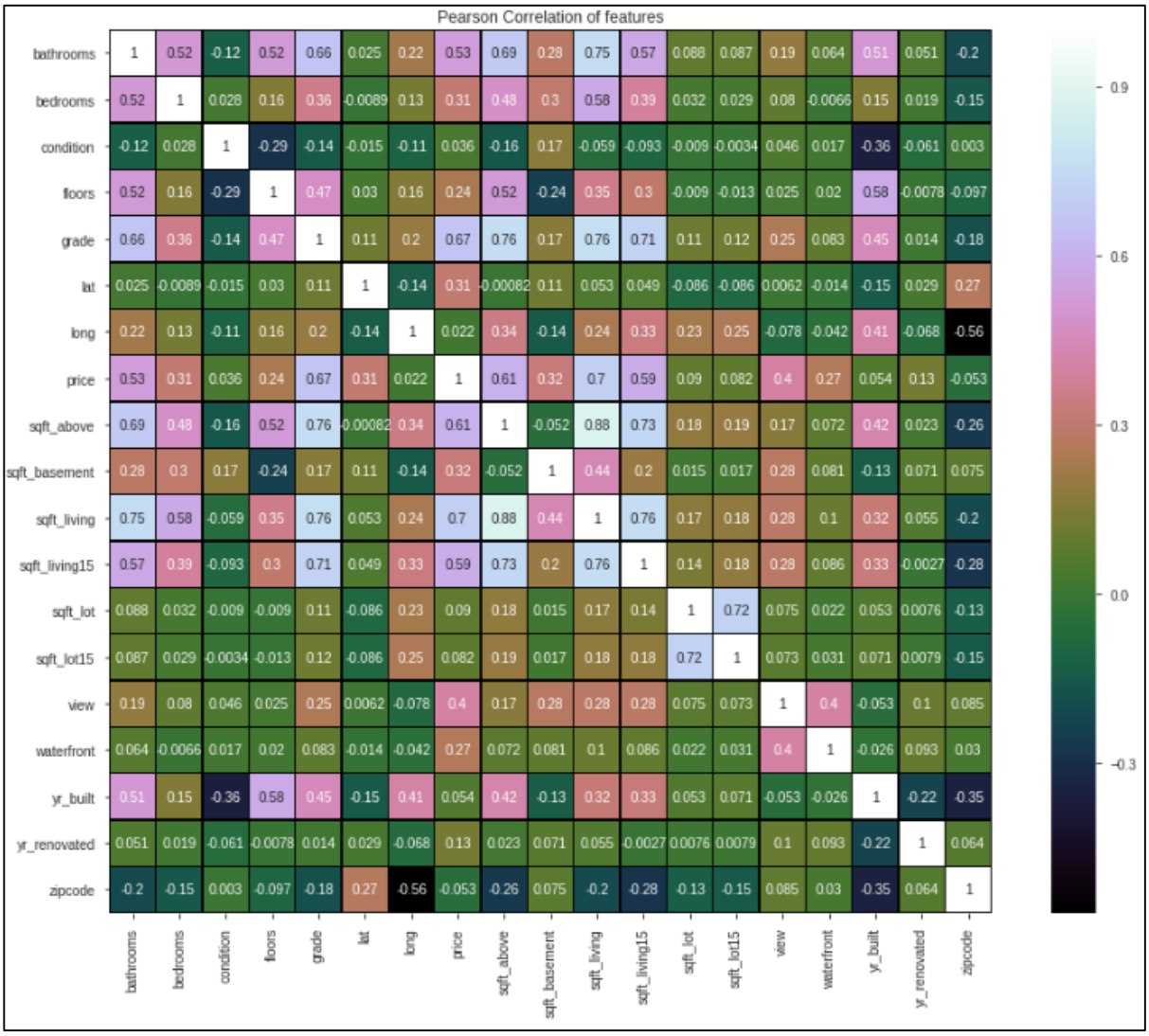

Рисунок 2.19 – Теплова кореляційна матриця ознак

З кореляційної матриці видно ступінь залежності між наступними ознаками:

– sqft_living & sqft_above (0.88);
– sqft_living15 & sqft_living (0.76);
– sqft_living & grade (0.76);
– sqft_above & grade (0.76);
– sqft_living & bathrooms (0.75);
– sqft_living15 & sqft_above (0.73);
– sqft_lot15 & sqft_lot (0.72);
– sqft_living15 & grade (0.71).

Побудуємо схему трьох гістограм: гістограма цін, графік розподілу цін, «коробка з вусами» цін продажу. Діаграма наведена на рисунку 2.20.

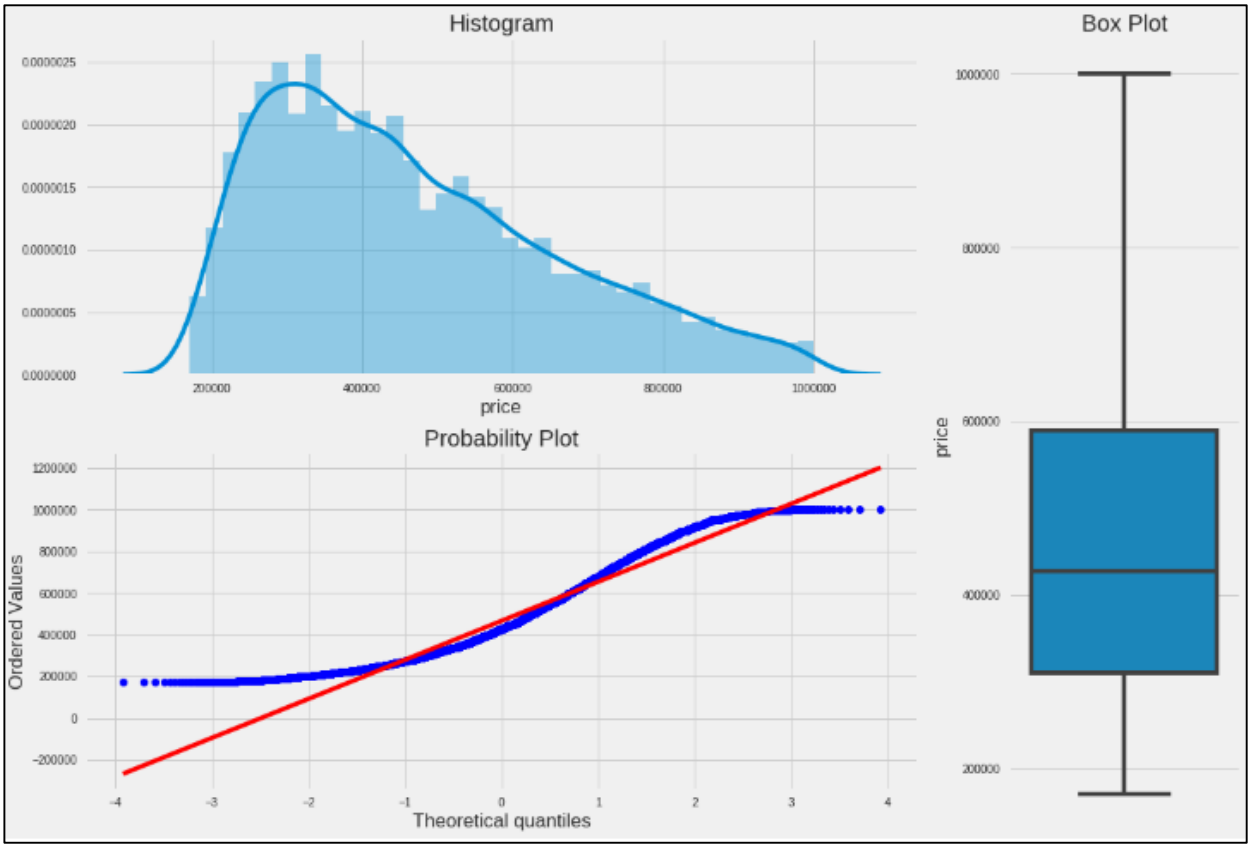

Рисунок 2.20 – Перевірка на аномальні дані методами Matplotlib, Pandas та Seaborn

З рисунка 2.20, а саме з гістограми цін можна побачити, що більша частина будинків має ціну від 200 тис. доларів до 450 тис. доларів, будинки з ціною вищою за 800 тис. доларів трапляються рідше, підтвердженням того є діаграма «коробка з вусами» з неї теж випливає, що середня ціна будинку варіюється між значеннями 300 тис. доларів та 600 тис. доларів.

З графіку ймовірностей видно, що наша лінія проходить через «синусоїду», майже 80% даних знаходяться навколо даної лінії. Чим більше даних охоплює лінія, тим кращим буде передбачення. Найточніше значення передбачення ціни продажу буде для будинків з ціною 200 тис. доларів та теоретичним квантилем -1, 600 тис. доларів, 1 млн доларів з теоретичним квантилем 3, тому, що саме на перетині цих значень проходить лінія передбачення.

Звідси можна зробити висновок, чим ближче находиться значення до лінії, тим точніший буде результат передбаченої ціни продажу.

Побудова та виведення діаграми щільності значень відповідно до проміжку ціни наведено на рисунках 2.21, 2.22.

```
y = np.array(train0.price)
plt.subplot(131)
plt.plot(range(len(y)),y,'.');plt.ylabel('price');plt.xlabel('index');
plt.subplot(132)
sns.boxplot(y=train0.price)
```

Рисунок 2.21 – Фрагмент коду для побудови діаграми щільності значень ціни продажу

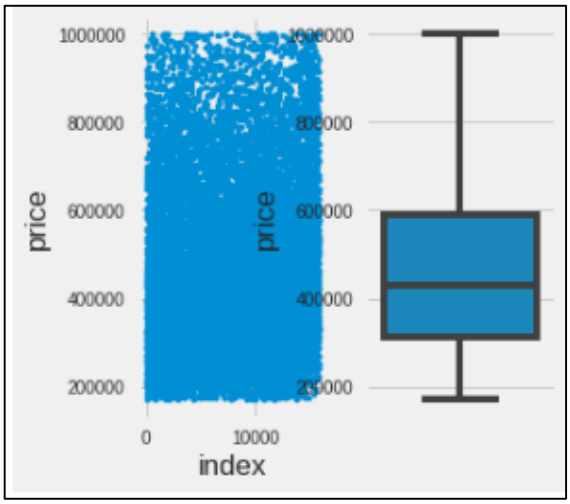

Рисунок 2.22 – Діаграма щільності значень ціни продажу

З рисунка 2.22 видно, що будинків вартість яких більша ніж 800 тис. доларів все менше і менше, аналогічна ситуація з будинками ціна яких менша за 200 тис. доларів, також видно що основна кількість будинків має вартість від 300 тис. доларів до 600 тис. доларів.

Побудова та виведення тривимірної діаграми значенням осі якої є: ціна, рік побудови та оцінка по 5-ти бальній шкалі наведено на рисунках 2.23, 2.24.

```
fig = plt.figure(figsize=(10,10))
ax = plt.axes(projection="3d")

z_points = train0['price']
x_points = train0['condition']
y_points = train0['yr_built']
ax.scatter3D(x_points, y_points, z_points, c=z_points, cmap='hsv');

ax.set_xlabel('condition')
ax.set_ylabel('yr_built')
ax.set_zlabel('price')

plt.show()
```

Рисунок 2.23 – Фрагмент коду для побудови тривимірної діаграми

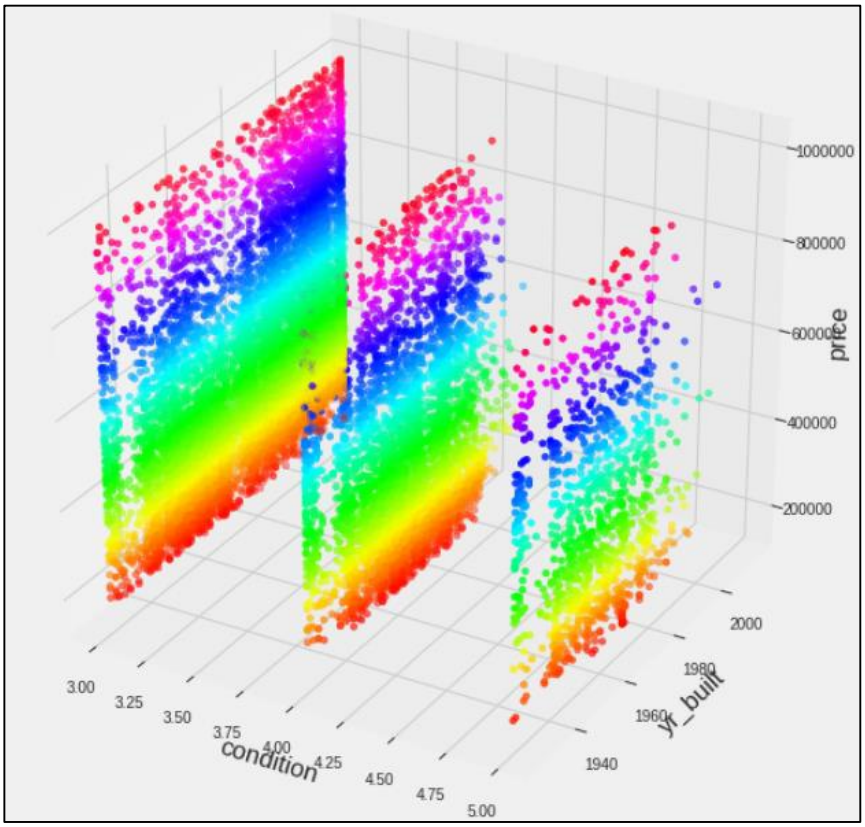

Рисунок 2.24 – Тривимірна діаграма (рік, ціна, оцінка)

З рисунка 2.24 видно, що переважна кількість будинків має оцінку 3 та 4 бали. Кількість будинків, що оцінені максимально спостерігається значно менше. Також можна спостерігати, що майже усі будинки, які побудовані в періоді 1930-тих роках відсутні.

Побудуємо тривимірну діаграму, на якій буде зображено площа будинку, кількість спальних кімнат, кількість поверхів. Побудова діаграми та її вигляд наведено на рисунках 2.25, 2.26.

```
fig=plt.figure(figsize=(19,12.5))
ax=fig.add_subplot(2,2,2, projection="3d")
ax.scatter(train0['floors'],train0['bedrooms'],train0['sqft_living'],c="darkgreen",alpha=.5)
ax.set(xlabel='\nFloors',ylabel='\nBedrooms',zlabel='\nsqft Living')
ax.set(ylim=[0,12])
plt.show()
```

Рисунок 2.25 – Фрагмент коду для побудови тривимірної діаграми

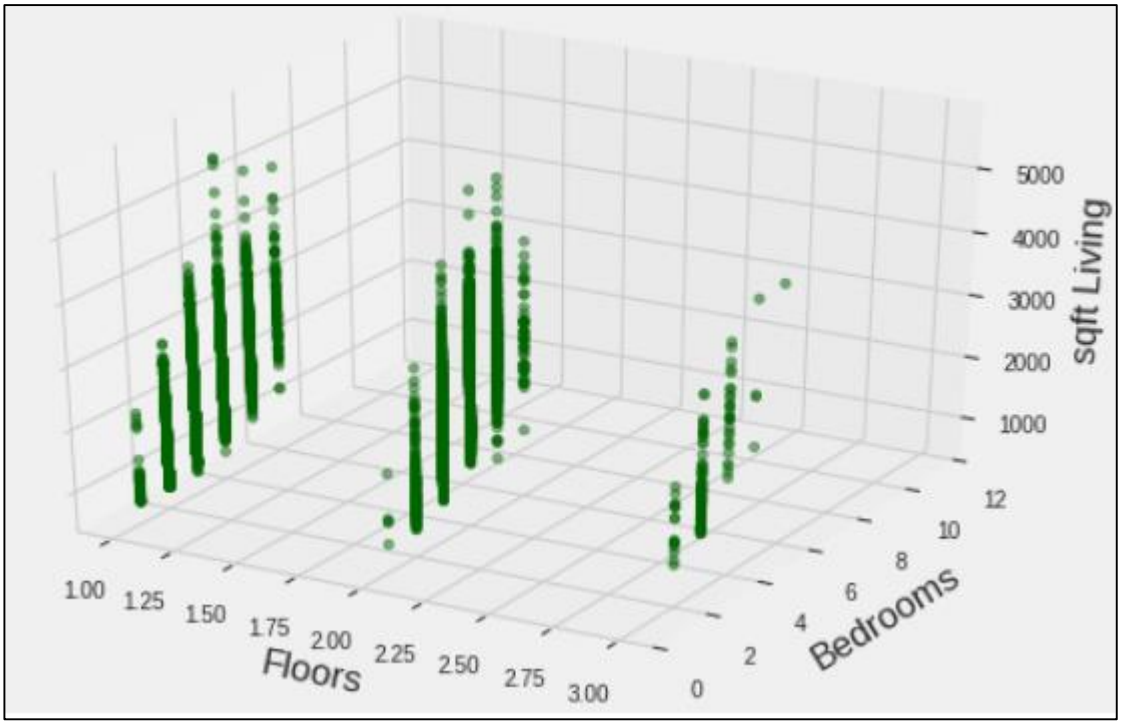

Рисунок 2.26 – Тривимірна діаграма (площа будинку, кількість спалень, кількість поверхів)

З рисунка 2.26 видно, що одноповерхові будинки з шести спальними кімнатами мають більшу площу ніж трьох поверхові будинки з шести спальнями кімнатами).

Побудуємо гістограму, на якій буде наведено ціни будинків відповідно до кількості спальних кімнат (рис. 2.27).

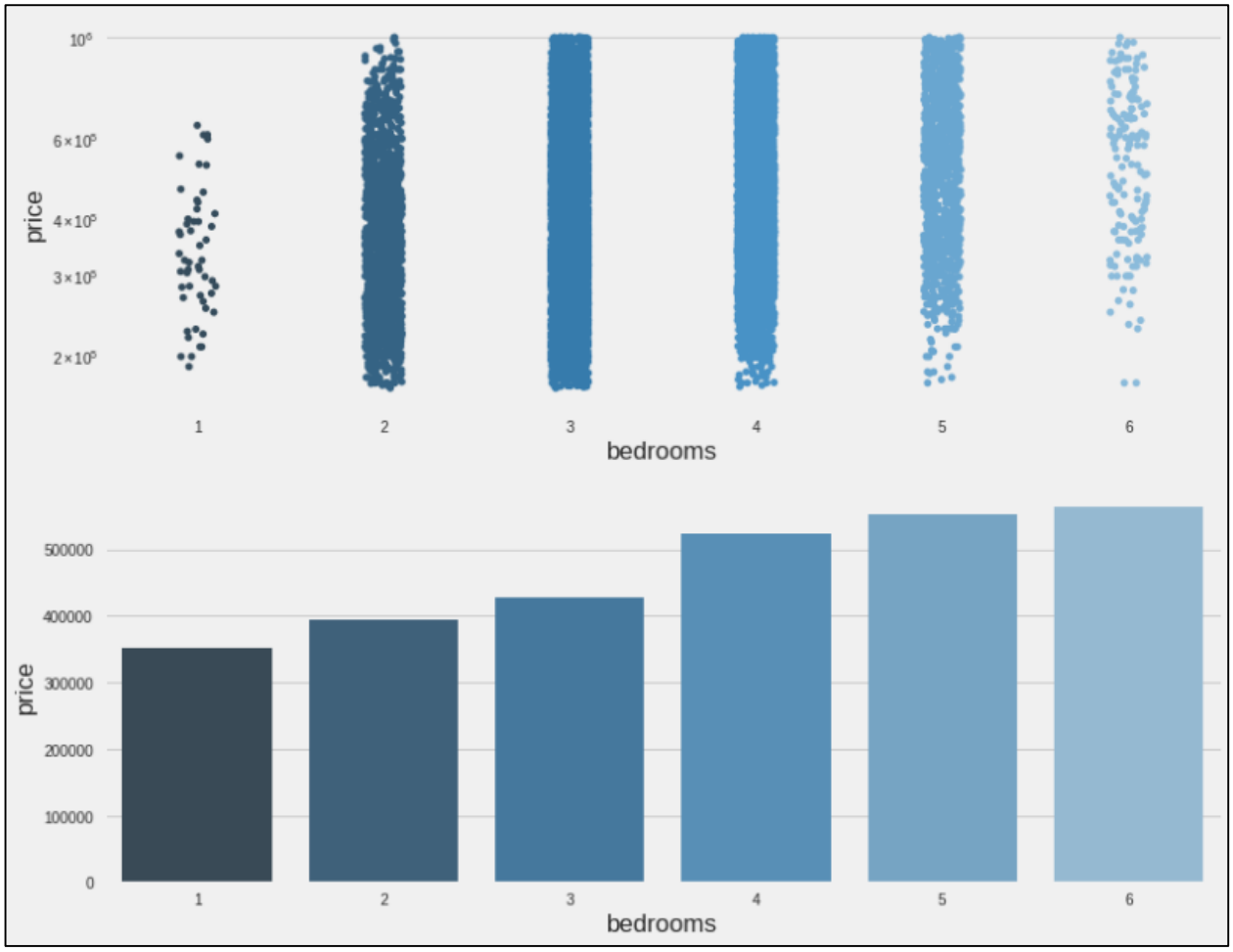

Рисунок 2.27 – Ціна на будинки відповідно до кількості спальних кімнат

З рисунка 2.27 бачимо, що більша кількість будинків має кількість спальних кімнат 2, 3 та 4, будинки, у яких 1 або 6 спальних кімнат зустрічаються рідше всього.

Будинки, які мають 5 або 6 спальних кімнат входять до однієї цінової категорії. Натомість спостерігається велика різниця у ціні, а саме 100 тис. доларів між будинками з 3 та 4 спальними кімнатами.

Виведемо у вигляді гістограми та візуалізації у вигляді «мішка слів» роки побудови будинків, які найбільше зустрічаються при продажі (рис. 2.28, 2.29).

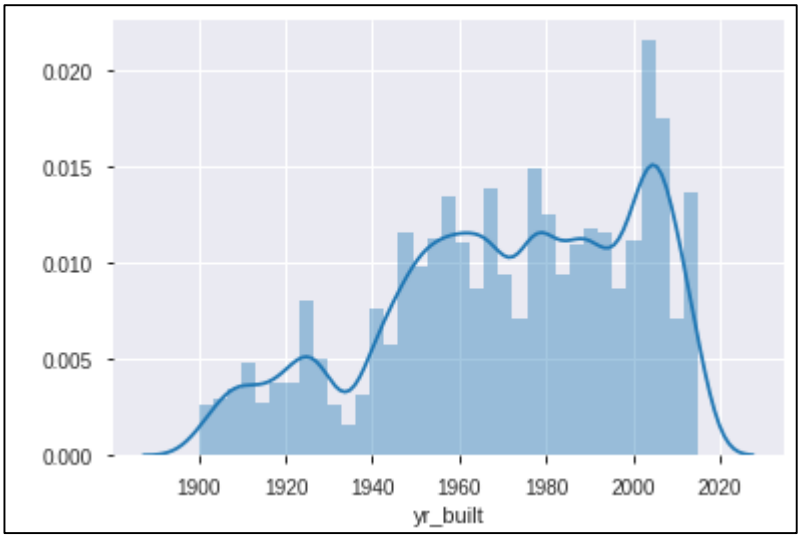

Рисунок 2.28 – Роки побудови будинків

З рисунка 2.28 видно, що більшість будинків побудовано у період 2000 – 2005 років, також є багато будинків 1968 – 1980 років побудови, будинки побудовані в інший період часу зустрічаються рідше.

Рисунок 2.29 – Візуалізація років побудови будинків

Більш чітко видно роки побудови будинків з візуалізації у вигляді «мішка зі слів», що на рисунку 2.29, де найбільшим розміром вказані роки побудови будинків, які зустрічаються найчастіше.

Маючи ознаки lat і long виведемо місця розташування кожного з будинків на мапу. Побудова та вигляд мапи наведено на рисунках 2.30 – 2.32.

```
houses_map = folium.Map(location = [train0['lat'].mean(), train0['long'].mean()], zoom_start =
10)
lat_long_data = train0[['lat', 'long']].values.tolist()
h_cluster = folium.plugins.FastMarkerCluster(lat_long_data).add_to(houses_map)
houses_map
```

Рисунок 2.30 – Побудова інтерактивної мапи розташування будинків

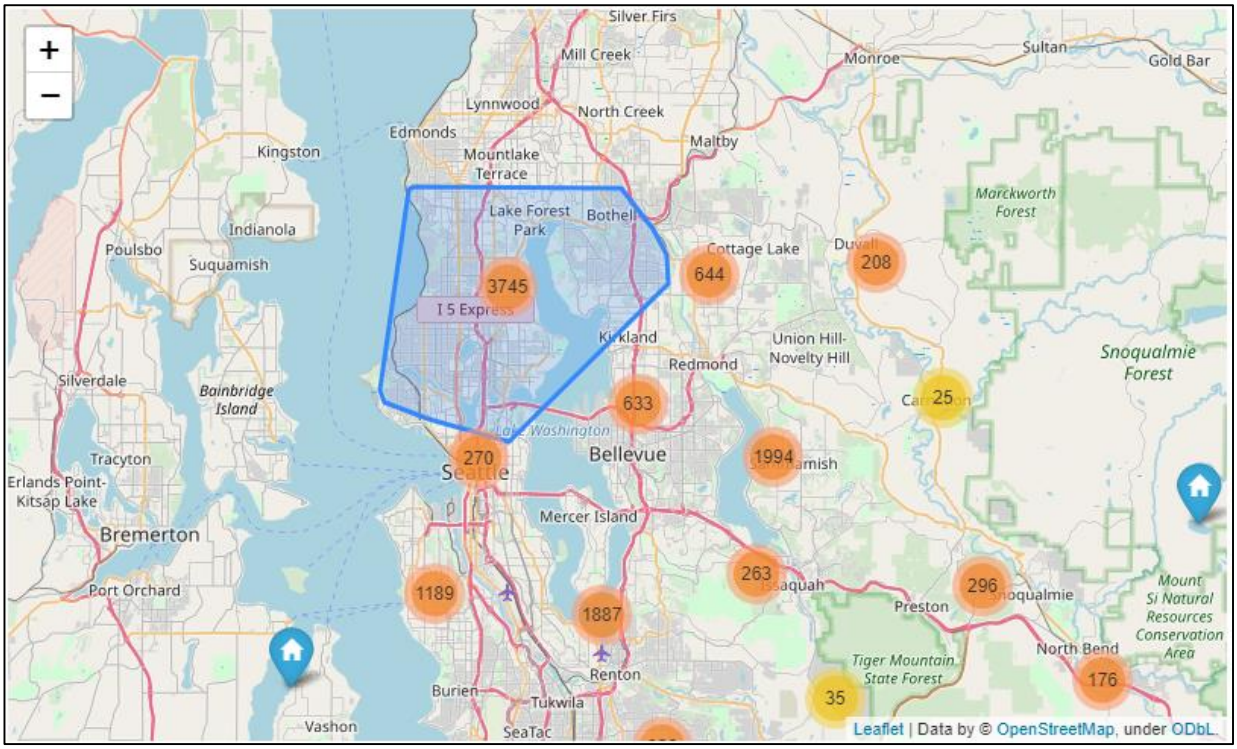

Рисунок 2.31 – Інтерактивна мапа розташування будинків

З рисунка 2.31 видно, що більшість будинків знаходиться у районі Lake Forest Park.

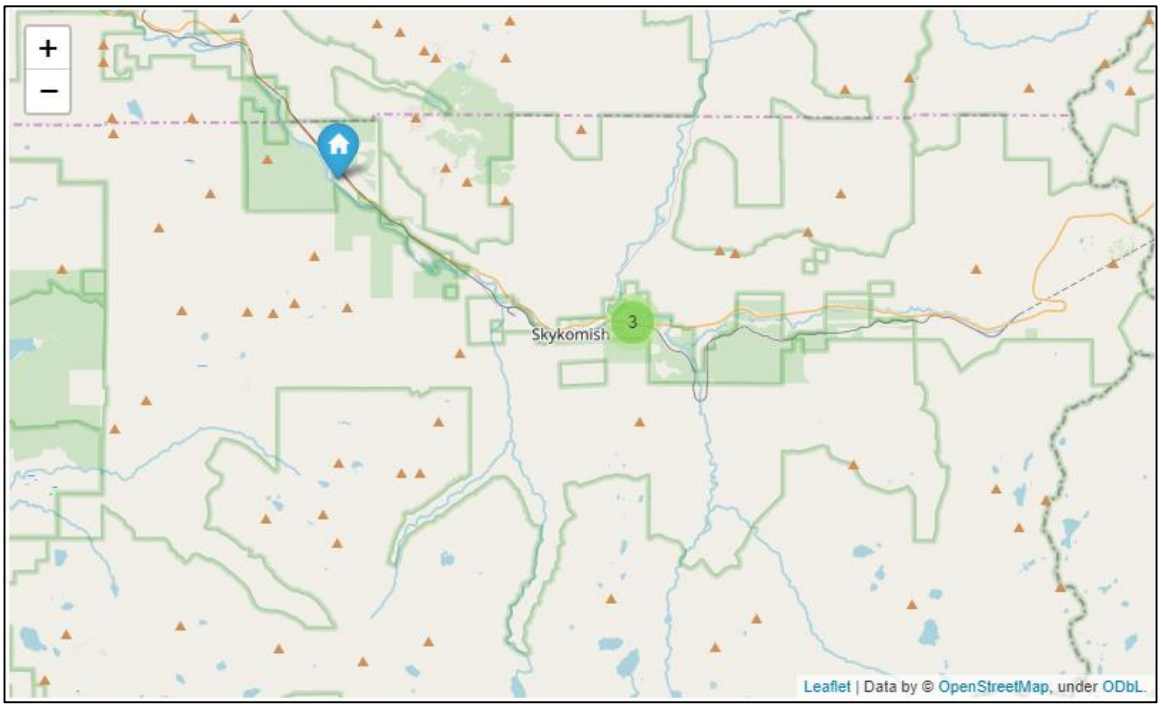

Рисунок 2.32 – Інтерактивна мапа розташування будинків з масштабом

З рисунка 2.32 видно, що найменша кількість будинків знаходиться саме у Skykomish.

Побудуємо інтерактивну кругову діаграму, яка відображає кількість відсотків будинків для кожної оцінки по 11 бальній шкалі оцінювання. Перевагою даного типу діаграми є те, що при наведенні на її «шматок» відображаються його характеристики. Побудова ти вигляд кругової діаграми наведено на рисунках 2.33, 2.34.

```
gradeframe = pd.DataFrame({"Grades":train0.grade.value_counts().index,"House_Grade":train0.grad
e.value_counts().values})
gradeframe["Grades"] = gradeframe["Grades"].apply(lambda x : "Grade " + str(x))
gradeframe.set_index("Grades",inplace=True)
p1 = [go.Pie(labels = gradeframe.index,values = gradeframe.House_Grade,hoverinfo="percent+label+
value",hole=0.1,marker=dict(line=dict(color="#000000",width=2)))]
layout4 = go.Layout(title="Grade Pie Chart")
fig4 = go.Figure(data=p1,layout=layout4)
iplot(fig4)
```

Рисунок 2.33 – Побудова інтерактивної кругової діаграми

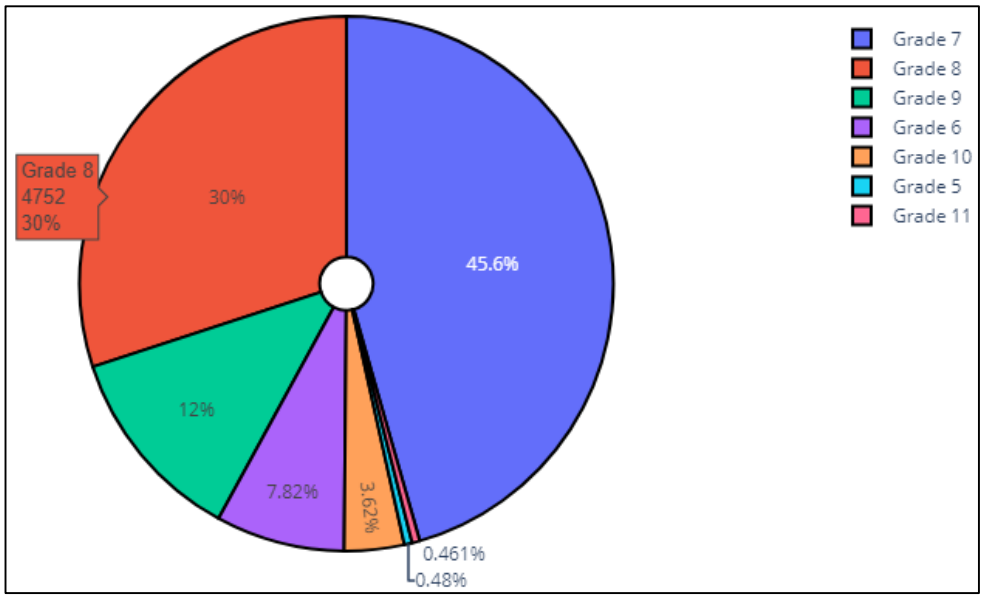

Рисунок 2.34 – Кругова діаграма оцінок

З рисунка 2.34 видно, що 45% серед усіх будинків, що оцінено на 7 балів з 11, 30 відсотків будинків мають оцінку 8, оцінку 5 та 11 має найменша кількість будинків.

Побудуємо інтерактивну кругову діаграму, яка відображатиме відсоток кількості будинків, відповідно до кількості спальних кімнат будинку. Діаграму наведено на рисунку 2.35.

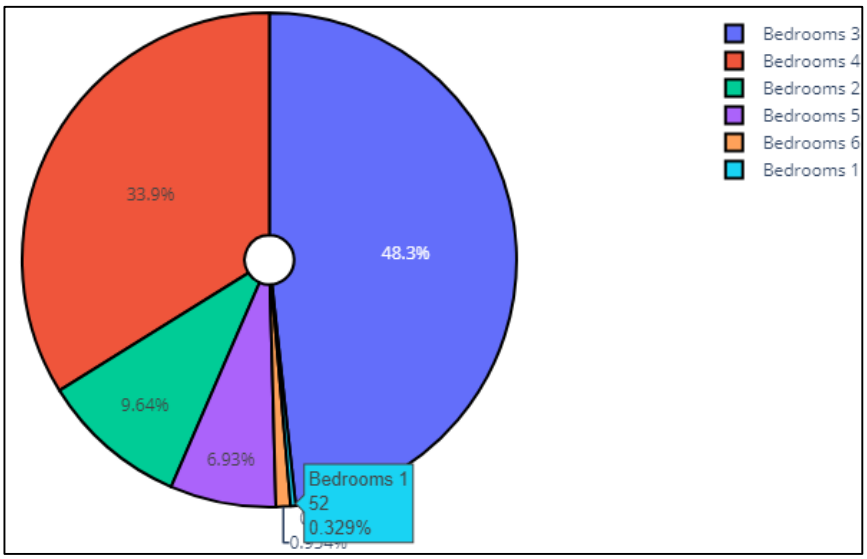

Рисунок 2.35 – Кругова діаграма (відсоток кількості будинків, відповідно до кількісті спальних кімнат)

З рисунка 2.35 видно, що найбільшим попитом на будинки є будинки, у яких 3 або 4 спальних кімнати та зрідка 2 або 5, як показує діаграма найменшим попитом є будинки, які мають лише одну спальну кімнату або 6.

Побудуємо інтерактивну кругову діаграму, яка відображатиме відсоток кількості будинків, відповідно до кількості поверхів будинку. Діаграму наведено на рисунку 2.36.

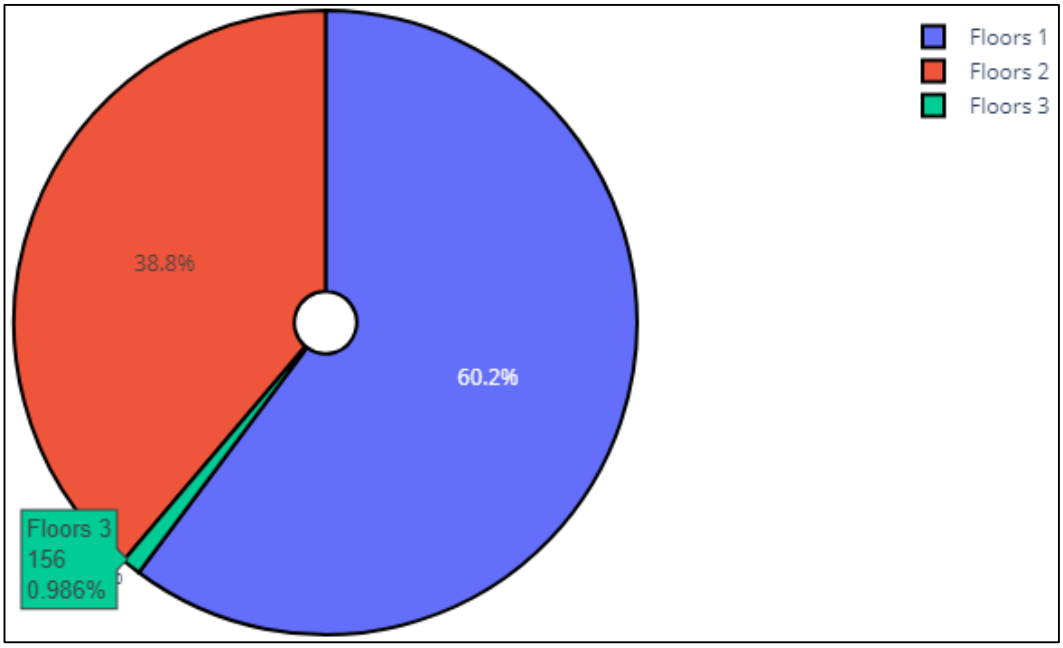

Рисунок 2.36 – Кругова діаграма (відсоток кількості будинків, відповідно до кількісті поверхів будинку)

З рисунка 2.36 видно, що найбільш кількість будинків, а саме 60% є одноповерховими, 38.8% будинків мають 2 поверхи, 3 поверхи мають приблизно 1% будинків.

Побудуємо інтерактивну кругову діаграму, яка відображатиме відсоток кількості будинків, відповідно до оцінки загального стану будинку. Діаграму наведено на рисунку 2.37.

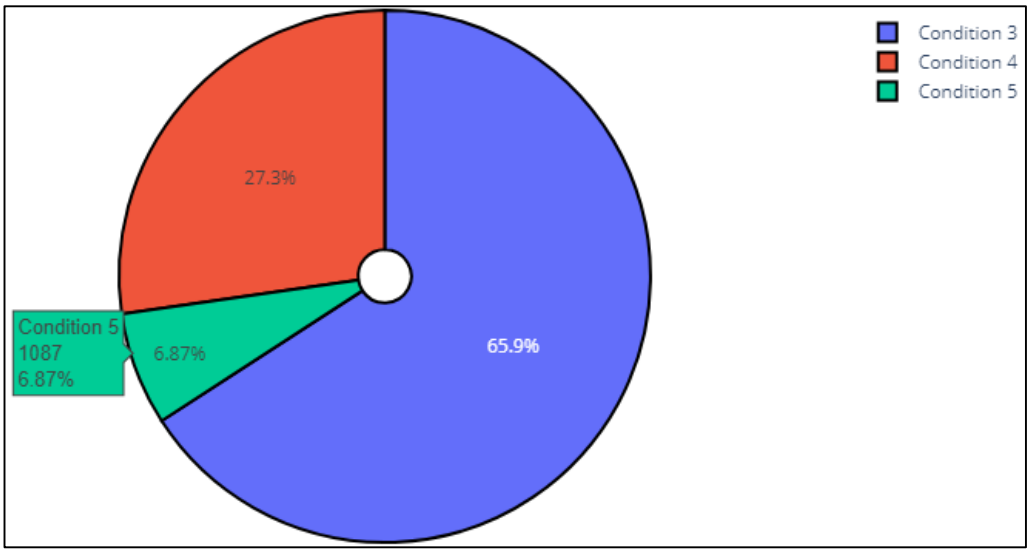

Рисунок 2.37 – Кругова діаграма (відсоток кількості будинків, відповідно до оцінки загального стану будинку)

З рисунка 2.37 видно, що стан більшості будинків (66%) оцінено на 3 бали з 5. Всього 6.9% будинків, які мають найкращий стан. В хорошому стані знаходиться 27.3% будинків.

За допомогою бібліотеки Pandas_profiling та її методу ProfileReport виведемо огляд нашого набору даних. Приклад використання наведено на рисунку 2.38

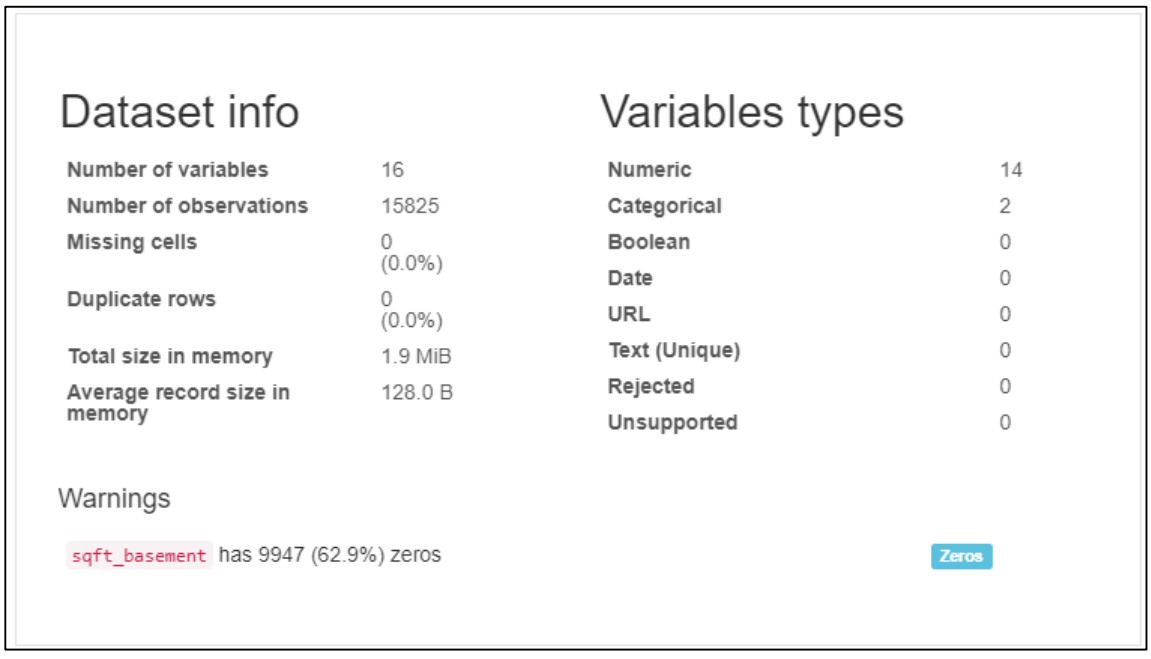

Рисунок 2.38 – Виведення інформації щодо огляду вхідного набору даних

З рисунка 2.38 видно, що серед 16 ознак є дві категоріальні ознаки, а саме condition та floors.

Використовуючи даний метод виведемо інформацію щодо основних статистичних даних декількох основних ознак (рис. 2.39 – 2.41).

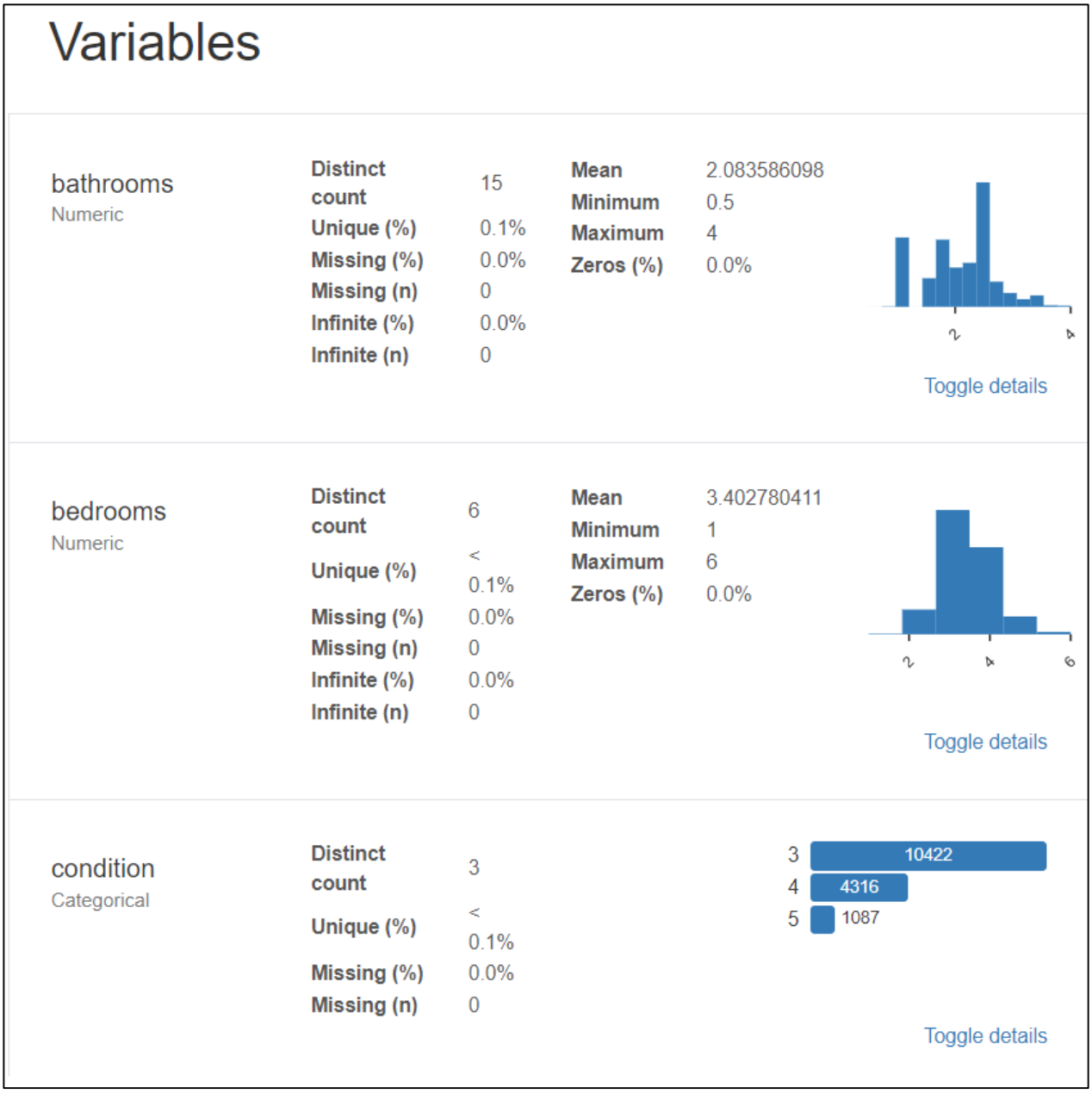

Рисунок 2.39 – Основні статистичні дані до кожної ознаки

Дивлячись на рисунок 2.39 видно, що середньостатистичний будинок має дві ванних кімнати та 3 спальних кімнат, будинки, які мають більше 4-рьох ванних кімнат не зустрічаються.

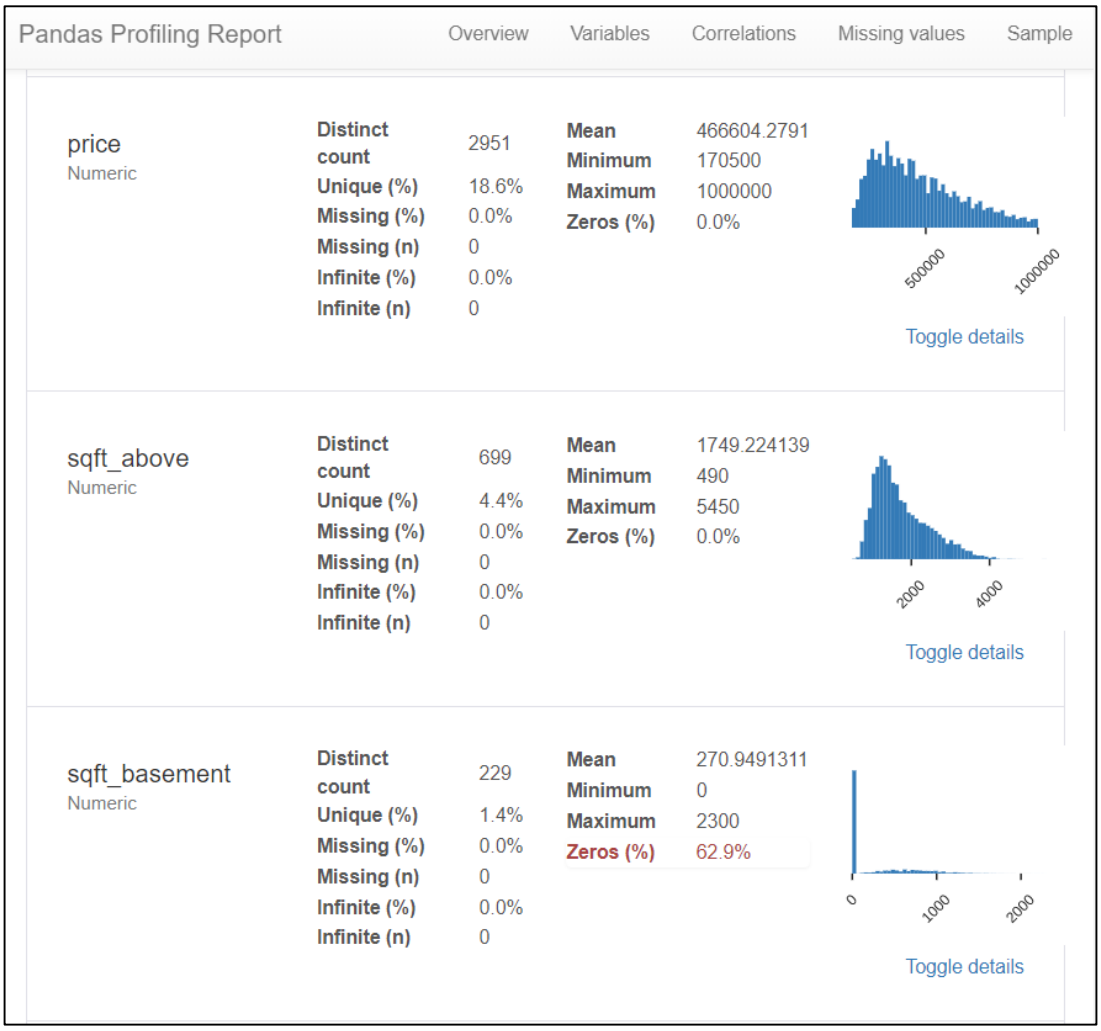

Рисунок 2.40 – Основні статистичні дані до ознак: ціна, площа будинку без підвального приміщення, площа підвального приміщення

З рисунка 2.40 бачимо, що середня ціна на будинок складає 466 тис. доларів, мінімальна ціна 170 тис. доларів, максимальна 1 мільйон доларів. Мінімальна площа будинку не враховуючи підвальне приміщення 490 квадратних футів (45.5 м^2). Середньостатистичний будинок має площу 1749 квадратних футів, тобто 162 квадратних метри. Максимальна площа будинку у даному наборі даних спостерігається 5450 квадратних футів (506 м^2).

Спостерігається, що більшість будинків не мають підвального приміщення, але середньостатистичний будинок має площу підвального приміщення 270 квадратних футів (25 м^2).

Рисунок 2.41 – Основні статистичні дані до вказаних ознак

З рисунка 2.41 видно, що площа середньостатистичного будинку враховуючи підвальне приміщення 2020 квадратних футів (187 м^2). Мінімальна площа будинку з підвальним приміщенням 710 квадратних футів (66 м^2). Найбільший будинок з підвальним приміщенням має площу 5635 квадратних футів (523 м^2).

Середньостатистична площа житлового приміщення найближчих 15 сусідів 1966 квадратних футів (182 м^2). Середньостатистична площа земельної ділянки 9652 квадратних футів (896 м^2). Найменша земельна ділянка має площу 1501 квадратних футів (139 м^2).

2.6 Висновки

У даному розділі здійснено вибір оптимальних інформаційних технологій та запропоновано алгоритм інформаційної технології передбачення ціни будинків, яка пропонується.

В якості мови програмування обрано мову Python, оскільки вона найбільше підходить для роботи з методами машинного навчання, в якості середовища розроблення програмного коду на Python обрано систему Kaggle IDE Notebooks.

Визначено, що задача передбачення ціни на будинки відноситься саме до виду машинного навчання з учителем (контрольоване навчання). А одним із найкращих варіантів розв’язку даної задачі є розв’язок за допомогою моделей регресії та моделей, які побудовані на основі дерев рішень.

Здійснено вибір та опис регресійних моделей машинного навчання, за допомогою яких відбувалось передбачення ціни продажу будинків, описано бібліотеки мови програмування Python, які використовуються для розв’язання поставленої задачі.

Виконано огляд та опис ознак вхідного набору даних та описано процес зчитування вхідного набору даних. Здійснено фільтрацію викидів та аномальних даних, визначено межі фільтрації та запропоновано правило фільтрування аномальних даних.

На етапі розвідувального аналізу даних побудовано діаграми, інтерактивні мапи та наведено різні візуалізації, які дозволили знайти в ознаках загальні закономірності, розподіли та аномалії.

3 ПОБУДОВА МОДЕЛЕЙ ТА ПЕРЕДБАЧЕННЯ ЦІНИ ПРОДАЖУ БУДИНКІВ

3.1 Дослідження важливості ознак

Першим етапом дослідження важливості ознак є вибір моделей (методів), за допомогою яких буде оцінюватись важливість ознак. У даній роботі важливості ознак буде оцінюватись за допомогою трьох регресійних моделей (методів), а саме:

– LGBM;

– XGB;

– Linear Regression.

Наступним кроком необхідно створити повну копію вхідного набору даних, тому що регресійні моделі потребують, щоб на вхід подавались саме стандартизовані дані.

Під стандартизацією даних мається на увазі таке перетворення даних, після якого кожна ознака матиме середнє значення 0 і дисперсію 1. У бібліотеці Scikit-Learn вже є готові для цього функції. Приклад «глибинного» копіювання даних наведено на рисунку 3.1.

```
train_fe = copy.deepcopy(train0)
target_fe = train_fe['price']
del train_fe['price']
```

Рисунок 3.1 – Створення копії вхідного набору даних

Ініціалізуємо змінні “X” та “y” вказавши, що змінна X – це наш набір даних без цільової ознаки (ціни продажу), y – це цільова ознака (рис. 3.2).

```
X = train_fe
z = target_fe
```

Рисунок 3.2 – Створення вхідних змінних для моделей, які побудовані на основі дерев рішень

Наступним етапом є створення тренувальної (80% значень) та тестової вибірки (20% значень). Створення наведено на рисунку 3.3.

```
Xtrain, Xval, Ztrain, Zval = train_test_split(X, z, test_size=0.2, random_state=0)
train_set = lgb.Dataset(Xtrain, Ztrain, silent=False)
valid_set = lgb.Dataset(Xval, Zval, silent=False)
```

Рисунок 3.3 – Лістинг команд для створення тренувальної та тестової вибірок даних

Налаштовуємо модель LGBM для виведення діаграми важливості ознак. Налаштовування параметрів моделі наведено на рисунку 3.4.

```
params = {
        'boosting_type':'gbdt',
        'objective': 'regression',
        'num_leaves': 31,
        'learning_rate': 0.05,
        'max_depth': -1,
        'subsample': 0.8,
        'bagging_fraction' : 1,
        'max_bin' : 5000 ,
        'bagging_freq': 20,
        'colsample_bytree': 0.6,
        'metric': 'rmse',
        'min_split_gain': 0.5,
        'min_child_weight': 1,
        'min_child_samples': 10,
        'scale_pos_weight':1,
        'zero_as_missing': True,
        'seed':0,
    }

modelL = lgb.train(params, train_set = train_set, num_boost_round=1000,
                   early_stopping_rounds=50,verbose_eval=10, valid_sets=valid_set)
```

Рисунок 3.4 – Налаштовування параметрі для LGBM

Побудова та виведення діаграми важливості ознак наведено на рисунках 3.5, 3.6.

```
fig =  plt.figure(figsize = (15,15))
axes = fig.add_subplot(111)
lgb.plot_importance(modelL,ax = axes,height = 0.5)
plt.show();plt.close()
```

Рисунок 3.5 – Набір команд для створення та виведення діаграми важливості ознак

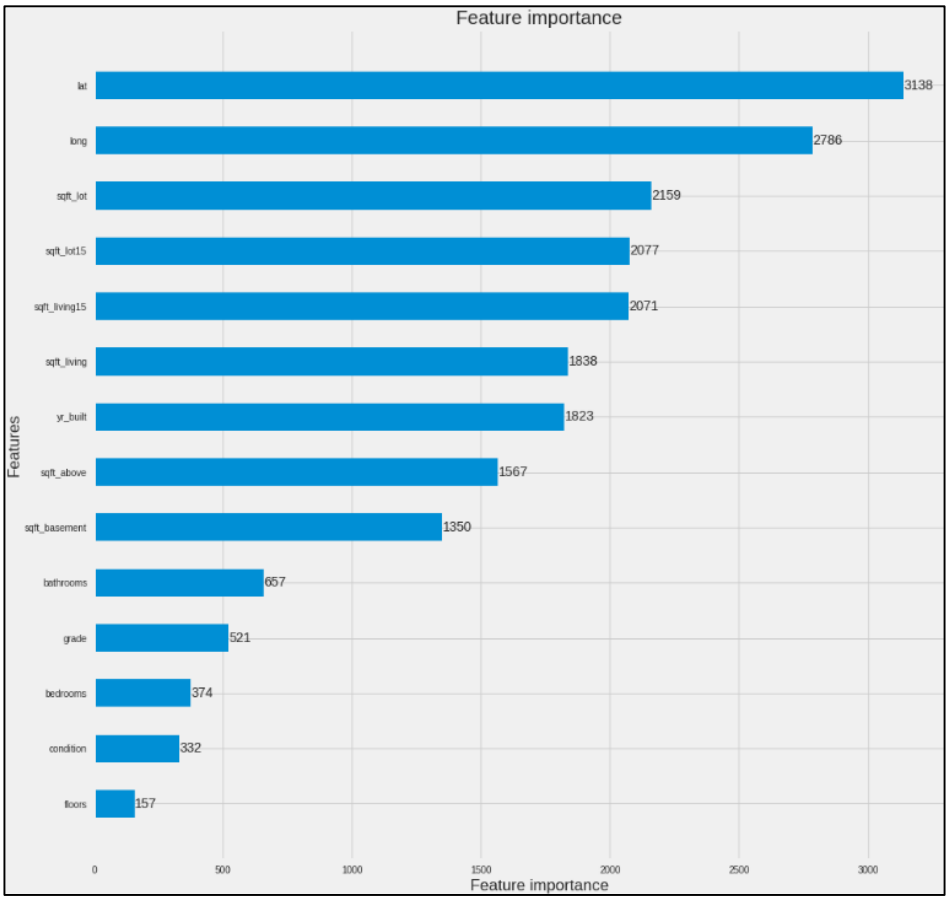

Рисунок 3.6 – Діаграма важливості ознак за моделлю LGBM

З рисунка 3.6 видно рівень важливості кожної ознаки, найважливішими виявились ознаки: lat, long, sqft_lot, sqft_lot15, sqft_living15, sqft_living, yr_built, sqft_above, sqft_basement. Такими ознаками, як bathrooms, grade, bedrooms, condition, floors можна знехтувати.

Діаграма важливості ознак за моделлю XGB наведено на рисунках 3.7, 3.8.

```
parms = {'max_depth':8, #maximum depth of a tree
         'objective':'reg:squarederror',
         'eta'      :0.3,
         'subsample':0.8,#SGD will use this percentage of data
         'lambda '  :4, #L2 regularization term,>1 more conservative
         'colsample_bytree ':0.9,
         'colsample_bylevel':1,
         'min_child_weight': 10}
modelx = xgb.train(parms, data_tr, num_boost_round=200, evals = evallist,
                   early_stopping_rounds=30, maximize=False,
                   verbose_eval=10)

print('score = %1.5f, n_boost_round =%d.'%(modelx.best_score,modelx.best_iteration))
```

Рисунок 3.7 – Налаштовування параметрів для моделі XGB

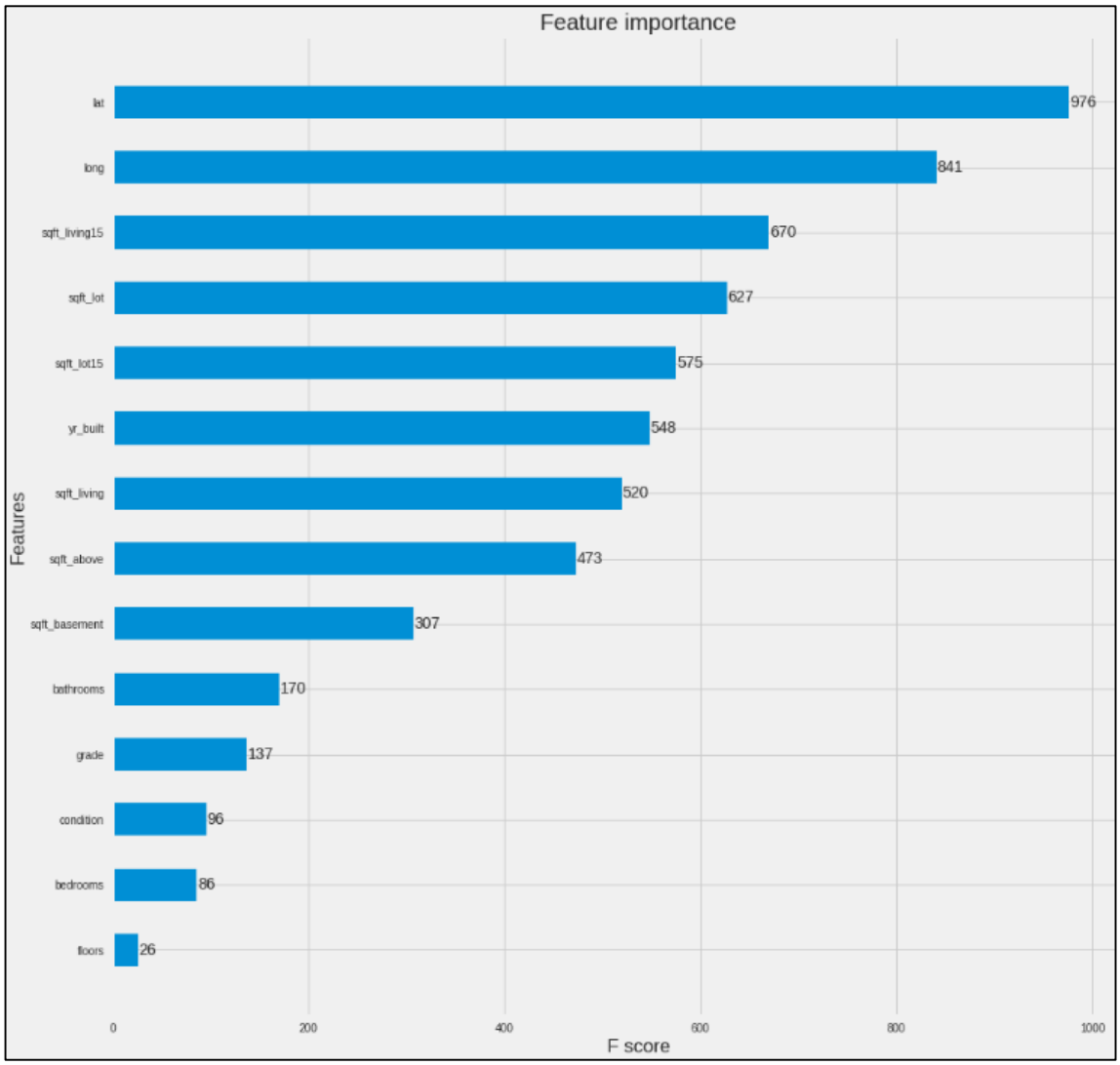

Рисунок 3.8 – Діаграма важливості ознак за моделлю XGB

З рисунка 3.8 видно важливість кожної ознаки, позиції важливості ознак за моделлю XGB та моделлю LGBM однакові, але значення оцінки незначною мірою відрізняються.

Стандартизація набору даних для регресійних моделей наведено на рисунку 3.9.

```
train_fe = pd.DataFrame(
    preprocessing.MinMaxScaler().fit_transform(train_fe),
    columns=train_fe.columns,
    index=train_fe.index
)
```

Рисунок 3.9 – Стандартизація тренувальної вибірки даних

Налаштування моделі лінійної регресії наведено на рисунку 3.10.

```
linreg = LinearRegression()
linreg.fit(train_fe, target_fe)
coeff_linreg = pd.DataFrame(train_fe.columns.delete(0))
coeff_linreg.columns = ['feature']
coeff_linreg["score_linreg"] = pd.Series(linreg.coef_)
coeff_linreg.sort_values(by='score_linreg', ascending=False)
```

Рисунок 3.10 – Налаштування лінійної регресії

Побудова та вигляд діаграми важливості ознак за трьома моделями, середнім та загальним значенням наведено на рисунках 3.11, 3.12.

```
feature_score['total'] = 0.5*feature_score['score_lgb'] + 0.3*feature_score['score_xgb'] \
                       + 0.2*feature_score['score_linreg']

# Plot the feature importances
feature_score.sort_values('total', ascending=False).plot(kind='bar', figsize=(20, 10))
```

Рисунок 3.11 – Побудова загальної діаграми важливості ознак

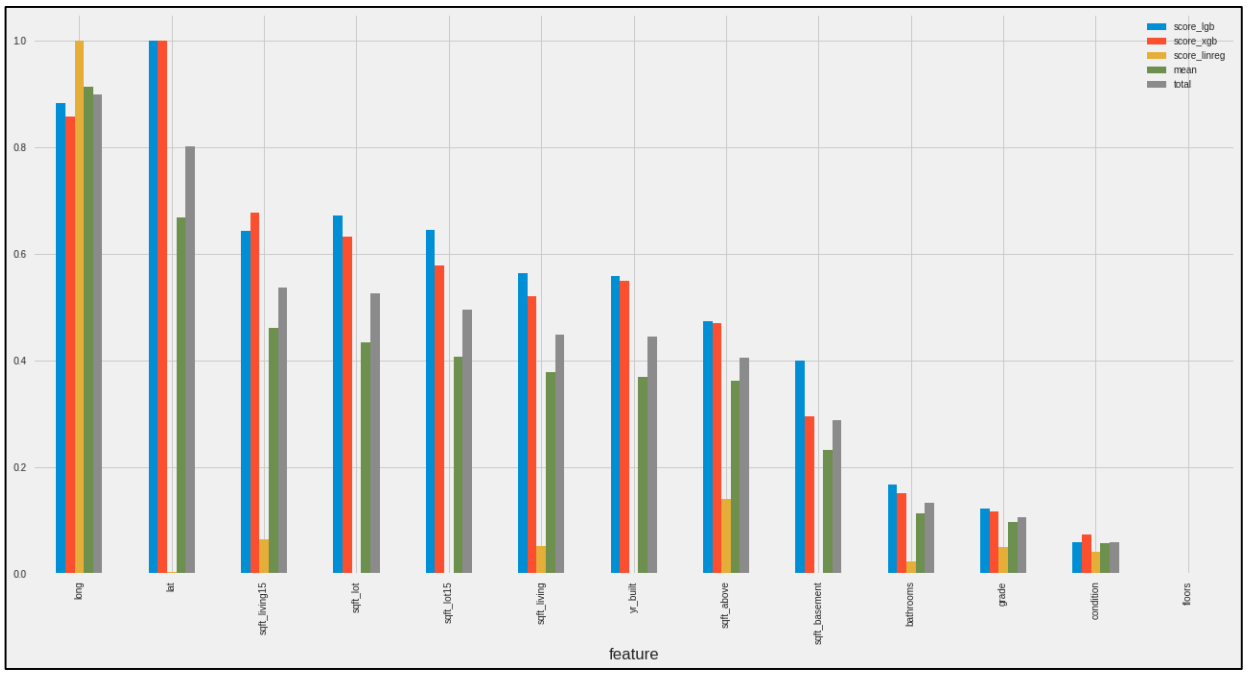

Рисунок 3.12 – Діаграма важливості ознак

Виведемо таблицю значень оцінки кожної ознаки, таблиця відсортована у порядку спадання за стовпцем «total» (рис. 3.13).

	score_lgb	score_xgb	score_linreg	mean	total
feature					
long	0.881919	0.857895	1.000000	0.913271	0.898328
lat	1.000000	1.000000	0.003846	0.667949	0.800769
sqft_living15	0.642066	0.677895	0.064826	0.461596	0.537367
sqft_lot	0.671587	0.632632	0.000095	0.434771	0.525602
sqft_lot15	0.644079	0.577895	0.000109	0.407361	0.495430
sqft_living	0.563905	0.520000	0.053051	0.378985	0.448563
yr_built	0.558873	0.549474	0.000046	0.369464	0.444288
sqft_above	0.472996	0.470526	0.139982	0.361168	0.405652
sqft_basement	0.400201	0.295789	0.000048	0.232013	0.288847
bathrooms	0.167729	0.151579	0.023281	0.114196	0.133994
grade	0.122107	0.116842	0.051267	0.096739	0.106359
condition	0.058705	0.073684	0.042376	0.058255	0.059933
floors	0.000000	0.000000	0.000000	0.000000	0.000000

Рисунок 3.13 – Коефіцієнт важливості кожної ознаки

З рисунка 3.13 по стовпцю «total» та рисунку 3.12 видно остаточний рівень важливості кожної ознаки. Якщо значення ознаки вище або рівне 0.50 можна вважати, що ці ознаки достатньо впливають на рівень точності передбачення, все що нижче цього значення має менший або не достатній рівень впливу.

Звідси можна виділити 7 найголовніших ознак, які впливають на передбачення ціни продажу будинку:

– lat;
– long;
– sqft_lot;
– sqft_living15;
– sqft_lot15;
– sqft_living;
– yr_built.

3.2 Підготовка вхідних даних для побудови моделей

Створюємо набір даних з відокремленим від нього цільовим значенням (рис. 3.14).

```
target_name = 'price'
train_target0 = train0[target_name]
train0 = train0.drop([target_name], axis=1)
```

Рисунок 3.14 – Лістинг команд для створення набору даних з відокремленим цільовим значенням

Розподіл даних на тестову та тренувальну вибірки наведено на рисунку 3.15.

```
train0, test0, train_target0, test_target0 = train_test_split(train0, train_target0, test_size=
0.2, random_state=0)
```

Рисунок 3.15 – Розподіл даних на тренувальну та тестову вибірки

Створення набору даних для моделей бустингу наведено на рисунку 3.16.

```
train0b = train0
train_target0b = train_target0
# Synthesis valid as test for selection models
trainb, testb, targetb, target_testb = train_test_split(train0b, train_target0b, test_size=vali
d_part, random_state=0)
```

Рисунок 3.16 – Вхідні дані для моделей бустингу

Створення вхідного стандартизованого набору даних для моделей, які використовують бібліотеку Sklearn наведено на рисунку 3.17.

```
scaler = StandardScaler()
train0 = pd.DataFrame(scaler.fit_transform(train0), columns = train0.columns)
```

Рисунок 3.17 – Стандартизований набір даних для моделей зі Sklearn

Тренувальна вибірка даних містить 8862 одиниць даних, тестова вибірка містить 3798 одиниць даних. Перед початком процесу моделювання підготовимо основні функції, які необхідні під час застосування моделі.

Ініціалізуємо порожні списки, які будуть містити значення метрик (оцінювання) (рис. 3.18).

```
acc_train_r2 = []
acc_test_r2 = []
acc_train_d = []
acc_test_d = []
acc_train_rmse = []
acc_test_rmse = []
```

Рисунок 3.18 – Ініціалізація порожніх списків, для значень метрики

Відносну похибку між передбаченим значенням y_pred та вимірюваними значеннями y_meas, а також RMSE (корінне середньоквадратичне відхилення) між передбаченим значенням y_pred та виміряними значеннями y_meas будемо розраховувати за допомогою створеної функції. Лістинг функції для обчислень наведено на рисунку 3.19.

```
def acc_d(y_meas, y_pred):
    # Relative error between predicted y_pred and measured y_meas values
    return mean_absolute_error(y_meas, y_pred)*len(y_meas)/sum(abs(y_meas))

def acc_rmse(y_meas, y_pred):
    # RMSE between predicted y_pred and measured y_meas values
    return (mean_squared_error(y_meas, y_pred))**0.5
```

Рисунок 3.19 – Функція для обчислення відносної похибки та RMSE

3.3 Вибір оптимальних налаштувань моделей

Для побудови моделей машинного навчання для задач з учителем найголовнішим завданням є уникнути перенавчання (overfitting), коли модель надто налаштована на навчальну вибірку, але потім показує погані результати на

інших (тестових) даних, хоча недонавчання (поганий збіг з навчальною вибіркою) також є проблемою [48].

Виконаємо налаштування таких моделей, як: Random Forest, XGB, LGBM, BaggingRegressor, ExtraTreesRegressor, Linear Regression, MLPRegressor.

Побудова та налаштовування моделі XGB і виведення значення точності та похибок наведено на рисунках 3.20, 3.21.

```
xgb_clf = xgb.XGBRegressor({'objective': 'reg:squarederror'})
parameters = {'n_estimators': [60, 100, 120, 140],
              'learning_rate': [0.01, 0.1],
              'max_depth': [5, 7],
              'reg_lambda': [0.5]}
xgb_reg = GridSearchCV(estimator=xgb_clf, param_grid=parameters, cv=5, n_jobs=-1).fit(trainb, t
argetb)
print("Best score: %0.3f" % xgb_reg.best_score_)
print("Best parameters set:", xgb_reg.best_params_)
acc_boosting_model(2,xgb_reg,trainb,testb)
```

Рисунок 3.20 – Налаштування моделі XGBoost

```
Best score: 0.864
Best parameters set: {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 120, 'reg_lambd
a': 0.5}
target =  [515000 870000 265900 837700 394000]
ytrain =  [543004.06 849082.9  252332.44 815293.44 391878.4 ]
acc(r2_score) for train = 0.957
acc(relative error) for train = 6.824
acc(rmse) for train = 39710.28
target_test = [554000 910000 403000 978500 528000]
ytest = [563729.25 888718.94 395501.6  870250.3  521896.3 ]
acc(r2_score) for test = 0.87
acc(relative error) for test = 10.811
acc(rmse) for test = 70146.096
```

Рисунок 3.21 – Результат навчання моделі XGBoost, значення точності та похибок

З рисунка 3.21 видно, що точність моделі XGBoost на тренувальній вибірці даних становить 0.95, на тестовій вибірці точність 0.869. Дані налаштування є оптимальними, перенавчання моделі не виявлено.

Побудова та налаштовування моделі LGBM і виведення значення точності та похибок наведено на рисунках 3.22, 3.23.

```
params = {
        'boosting_type':'gbdt',
        'objective': 'regression',
        'num_leaves': 31,
        'learning_rate': 0.01,
        'max_depth': -1,
        'subsample': 0.8,
        'bagging_fraction' : 1,
        'max_bin' : 5000 ,
        'bagging_freq': 20,
        'colsample_bytree': 0.6,
        'metric': 'rmse',
        'min_split_gain': 0.5,
        'min_child_weight': 1,
        'min_child_samples': 10,
        'scale_pos_weight':1,
        'zero_as_missing': False,
        'seed':0,
    }
modelL = lgb.train(params, train_set = train_set, num_boost_round=10000,
                   early_stopping_rounds=8000,verbose_eval=500, valid_sets=valid_set)
```

Рисунок 3.22 – Налаштовування моделі LGBM

```
target =  [515000 870000 265900 837700 394000]
ytrain =  [537547.04954839 854973.89112175 257594.7942209  815556.1951764
 381224.50945005]
acc(r2_score) for train = 0.944
acc(relative error) for train = 7.433
acc(rmse) for train = 45272.831
target_test = [554000 910000 403000 978500 528000]
ytest = [570607.60671493 903853.90159535 423494.04728682 859922.93835679
 551383.08592441]
acc(r2_score) for test = 0.876
acc(relative error) for test = 10.64
acc(rmse) for test = 68477.002
```

Рисунок 3.23 – Результат навчання моделі LGBM, значення точності та похибок

З рисунка 3.23 видно, що точність моделі LGBM на тренувальній вибірці даних становить 0.944, на тестовій вибірці точність – 0.876. Перенавчання моделі не спостерігається, налаштування моделі є оптимальними.

Побудова моделі «Random Forest» та результат навчання наведено на рисунку 3.24

```
target =  [515000 870000 265900 837700 394000]
ytrain =  [525390. 847990. 249250. 840160. 382495.]
acc(r2_score) for train = 0.971
acc(relative error) for train = 4.739
acc(rmse) for train = 32595.44
target_test = [554000 910000 403000 978500 528000]
ytest = [563220.  861900.  358045.  793521.1 535440. ]
acc(r2_score) for test = 0.844
acc(relative error) for test = 11.912
acc(rmse) for test = 76916.28
```

Рисунок 3.24 – Результат навчання моделі «Random Forest», значення точності та похибок

З рисунка 3.24 видно, що навчання моделі відбувається без вказаних додаткових параметрів навчання, тобто усі параметри використовуються за замовчуванням. На тренувальній вибірці точність моделі – 0.97, на тестовій – 0.844.

Побудова моделі «BaggingRegressor» та результат навчання наведено на рисунках 3.25, 3.26.

```
bagging = BaggingRegressor()
bagging.fit(train, target)
acc_model(3,bagging,train,test)
```

Рисунок 3.25 – Побудова моделі «BaggingRegressor» з параметрами налаштування за замовченням

```
target =  [515000 870000 265900 837700 394000]
ytrain =  [546615. 841990. 254455. 808820. 418050.]
acc(r2_score) for train = 0.971
acc(relative error) for train = 4.836
acc(rmse) for train = 32426.27
target_test = [554000 910000 403000 978500 528000]
ytest = [553200. 848440. 363950. 822800. 564180.]
acc(r2_score) for test = 0.845
acc(relative error) for test = 11.915
acc(rmse) for test = 76639.934
```

Рисунок 3.26 – Результат навчання моделі «BaggingRegressor», значення точності та похибок

Точність на тренувальному наборі даних – 0.97, на тестовому – 0.845.

Побудова моделі «ExtraTreesRegressor» та результат навчання наведено на рисунку 3.27.

```
target =  [515000 870000 265900 837700 394000]
ytrain =  [515000. 870000. 265900. 837700. 394000.]
acc(r2_score) for train = 0.998
acc(relative error) for train = 0.15
acc(rmse) for train = 8053.41
target_test = [554000 910000 403000 978500 528000]
ytest = [572090. 864240. 481950. 866100. 521170.]
acc(r2_score) for test = 0.835
acc(relative error) for test = 12.192
acc(rmse) for test = 78946.955
```

Рисунок 3.27 – Результат навчання моделі «ExtraTreesRegressor», значення точності та похибок

Точність на тренувальному наборі даних – 0.99, що являється майже максимальною, але на тестовому наборі даних точність – 0.83. Налаштування моделі вказано за замовчуванням. Спостерігається перенавчання.

Побудова моделі лінійної регресії та результат навчання наведено на рисунку 3.28.

```
target =  [515000 870000 265900 837700 394000]
ytrain =  [532822.70982975 569708.51702279 257564.67925537 803412.92766833
 330417.32661198]
acc(r2_score) for train = 0.677
acc(relative error) for train = 19.042
acc(rmse) for train = 108957.85
target_test = [554000 910000 403000 978500 528000]
ytest = [510358.16230839 954647.33826122 501134.68272856 747458.50282714
 472726.42348938]
acc(r2_score) for test = 0.673
acc(relative error) for test = 19.32
acc(rmse) for test = 111240.45
```

Рисунок 3.28 – Результат навчання моделі лінійної регресії, значення точності та похибок

Побудова моделі «MLPRegressor» та результат навчання наведено на рисунках 3.29, 3.30.

```
mlp = MLPRegressor()
param_grid = {'hidden_layer_sizes': [i for i in range(2,20)],
              'activation': ['relu'],
              'solver': ['adam'],
              'learning_rate': ['constant'],
              'learning_rate_init': [0.01],
              'power_t': [0.5],
              'alpha': [0.0001],
              'max_iter': [1000],
              'early_stopping': [True],
              'warm_start': [False]}
mlp_GS = GridSearchCV(mlp, param_grid=param_grid,
                      cv=10, verbose=True, pre_dispatch='2*n_jobs')
mlp_GS.fit(train, target)
acc_model(6,mlp_GS,train,test)
```

Рисунок 3.29 – Параметри налаштовування моделі «MLPRegressor»

```
target =  [515000 870000 265900 837700 394000]
ytrain =  [540189.81401968 603948.9764792  246037.96671905 827925.93250755
 361181.7550867 ]
acc(r2_score) for train = 0.698
acc(relative error) for train = 17.724
acc(rmse) for train = 105393.84
target_test = [554000 910000 403000 978500 528000]
ytest = [504724.76906366 920670.7789767  491122.57336574 760800.4029958
 476872.2475492 ]
acc(r2_score) for test = 0.693
acc(relative error) for test = 18.181
acc(rmse) for test = 107848.41
```

Рисунок 3.30 – Результат навчання моделі «MLPRegressor», значення точності та похибок

Точність моделі на тренувальному наборі даних 0.69, на тестовому наборі 0.69. Модель недостатньо охопила тенденцію вхідних даних, тому результат не дуже високий.

3.4 Оцінювання точності передбачення та порівняння моделей

Створення датафрейму, який містить усі необхідні результуючі значення навчання моделі наведено на рисунку 3.31. Узагальнене значення метрик для кожної побудованої моделі наведено на рисунку 3.32.

```
models = pd.DataFrame({
    'Model': ['Random Forest', 'XGB', 'LGBM', 'BaggingRegressor', 'ExtraTreesRegressor', 'Linear
Regression', 'MLPRegressor'],

    'r2_train': acc_train_r2,
    'r2_test': acc_test_r2,
    'd_train': acc_train_d,
    'd_test': acc_test_d,
    'rmse_train': acc_train_rmse,
    'rmse_test': acc_test_rmse
                    })
```

Рисунок 3.31 – Створення датафрейму значень оцінки та похибок моделей

	Model	r2_train	r2_test	d_train	d_test	rmse_train	rmse_test
2	LGBM	0.944	0.876	7.433	10.640	45,272.831	68,477.002
1	XGB	0.957	0.870	6.824	10.811	39,710.280	70,146.096
3	BaggingRegressor	0.971	0.845	4.836	11.915	32,426.270	76,639.934
0	Random Forest	0.971	0.844	4.739	11.912	32,595.440	76,916.280
4	ExtraTreesRegressor	0.998	0.835	0.150	12.192	8,053.410	78,946.955
6	MLPRegressor	0.698	0.693	17.724	18.181	105,393.840	107,848.410
5	Linear Regression	0.677	0.673	19.042	19.320	108,957.850	111,240.450

Рисунок 3.32 – Значення оцінки та похибок моделей

Графік точності моделей на тренувальній та тестовій вибірці за коефіцієнтом детермінації наведено на рисунку 3.33.

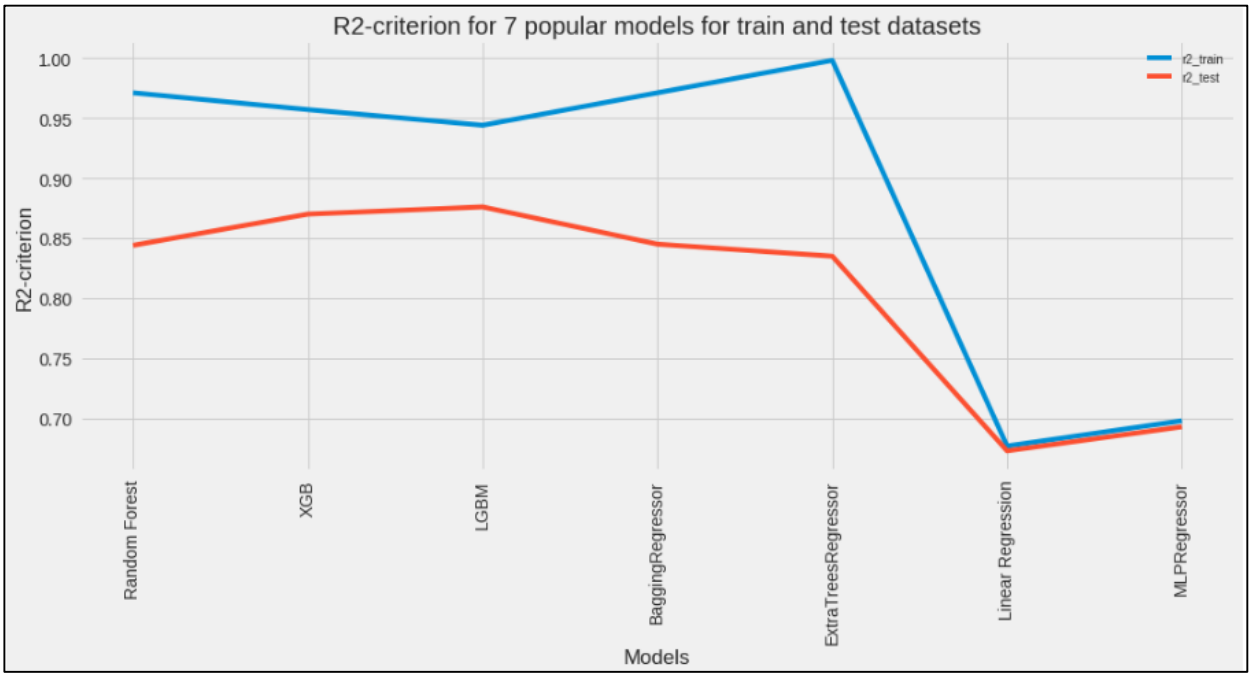

Рисунок 3.33 – Графік точності моделей

З рисунка 3.32 видно, що оптимальною моделлю є LGBM її застосування на тестовій вибірці дозволило передбачати ціни на будинки з точністю 0.876

Графік відносних похибок моделей між прогнозованими значеннями та виміряними значеннями наведена на рисунку 3.34.

Рисунок 3.34 – Графік відносних похибок моделей

З рисунка 3.34 видно, що найменшу похибку на тестових даних показує модель LGBM зі значенням 10.64.

Графік середньоквадратичних похибок моделей (RMSE) наведено на рисунку 3.35.

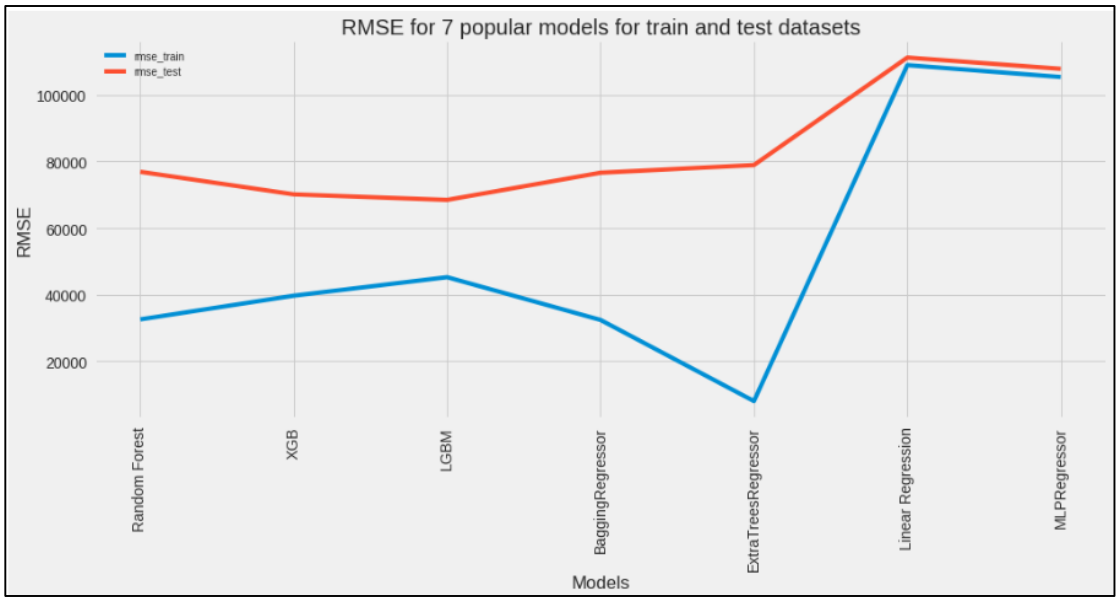

Рисунок 3.35 – Графік середньоквадратичної похибки моделей

З рисунка 3.35 видно, що модель LGBM має найменше значення похибки RMSE на тестовій вибірці даних, дана модель є оптимальною.

3.5 Передбачення значення цільової величини

Оголошення тестової вибірки для передбачення цільового значення наведено на рисунку 3.36.

```
testn = pd.DataFrame(scaler.transform(test0), columns = test0.columns)
```

Рисунок 3.36 – Тестова вибірка для передбачення

Передбачення цільового значення та фрагмент виведення передбачених значень за оптимальною моделлю LGBM наведено на рисунку 3.37.

```
# LGB Regression model for basic train
lgb_predict = modelL.predict(test0)
lgb_predict[:3]
```

```
array([405843.41265359, 381263.49625537, 302643.99607432])
```

Рисунок 3.37 – Передбачення цільового значення за моделлю LGBM

З рисунка 3.37 видно отримання набору даних, який містить передбачені значення ціни на будинки.

Виведемо фрагмент значень порівнянь між реальною ціною та передбаченим значенням ціни, різницею між ними, використовуючи модель LGBM, оскільки саме дана модель показала найкращу оцінку передбачення на тестовій вибірці. Виведення порівняльного фрагмента наведено на рисунку 3.38.

	Real_price	predicted_prices	difference
0	424000	405843	18157
1	520000	381263	138737
2	300000	302643	2643
3	275000	263021	11979
4	738950	707175	31775
5	257000	269355	12355
6	528000	463689	64311
7	316000	326276	10276
8	520000	507827	12173
9	615000	538111	76889
10	435000	460696	25696
11	959900	833830	126070
12	750000	582951	167049
13	364500	415817	51317
14	425000	410763	14237
15	490000	524937	34937
16	323000	336675	13675
17	475000	510345	35345
18	920000	749382	170618
19	456000	544712	88712

Рисунок 3.38 – Фрагмент результату передбачення та різницею між дійсною ціною

Усереднені значення по кожному стовпці наведено на рисунку 3.39.

	Name	Average
0	real_price	466,159.00
1	predicted_prices	466,751.00
2	difference	47,031.00

Рисунок 3.39 – Усереднені значення

Більш детально роботу можна переглянути у системі Kaggle, ноутбук «House Price Prediction by 7 models ML | EDA & FE» [49].

3.6 Висновки

Отже, у третьому розділі досліджено важливість ознак, побудовано діаграму важливості ознак за трьома моделями машинного навчання: LGBM, XGB, Linear regression. Визначено 5 головних ознак, які мають вплив на передбачення ціни продажу будинків:

– координати розташування будинку;
– площа земельної ділянки;
– площа житлового приміщення найближчих 15 сусідів;
– площа земельних ділянок найближчих 15 сусідів;
– рік побудови.

Підготовлено вхідні дані для побудови моделей, створено функцію для обчислення точності та похибок моделей.

Побудовано та налаштовано 7 моделей машинного навчання:

– Random Forest;
– XGB;
– LGBM;
– BaggingRegressor;
– ExtraTreesRegressor;
– Linear Regression;
– MLPRegressor.

Виконано передбачення ціни продажу будинків та порівняно моделі за точністю та похибками. Визначено оптимальну модель машинного навчання LGBM, точність передбачення якої на тренувальному наборі даних – 0.95, а на тестовому наборі даних – 0.876. Отже, що є більшим за 0.82, як у найкращого аналога (чим ближче метрика точності до 1, тим краще).

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нової інформаційної технології передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання. Особливістю розробки є підвищення точності передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання шляхом створення інформаційної технології передбачення цієї ціни.

За аналог було взято калькулятор розрахунку вартості нерухомості від української системи DOM.RIA. Орієнтовна вартість аналога для комерційного сегменту складає близько 8000$, тобто 320000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів Вінницького національного технічного університету кафедри системного аналізу та інформаційних технологій. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1 [50].

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах

Продовження табл. 4.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

Практична здійсненність					
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Бурденюк А. В.	Жук В. Л.	Гусак С. В.
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	3	4
Сума	44	42	44
Середньоарифметична сума балів	(44+42+44) / 3 = 43,33		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що оцінка вартості будинку – послуга, без якої не обійтись у багатьох випадках. Фактично будь-які операції з нерухомим майном вимагають розрахунку його ринкової вартості. Високий рівень комерційного потенціалу розроблюваного нового програмного продукту досягається за рахунок того, що використовуються методи машинного навчання, які дозволяють підвищити точність такого передбачення у порівнянні з аналогами.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{М}{Т_р} \cdot t, \quad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;
$Т_р$ – число робочих днів за місяць, 20 днів;
t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	53000	2650,00	38	100700,000
Програміст	46700	2335,00	38	88730,000
Всього				189430,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників, які брали участь в розробці програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 10 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 10\ \% / 100\ \% \qquad (4.2)$$

$$З_д = (189430{,}00 \cdot 10\ \% / 100\ \%) = 18943{,}00 \text{ (грн)}.$$

Нарахування на заробітну плату розробників. Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \qquad (4.3)$$

$$Н_з = (189430{,}00 + 18943{,}00) \cdot 22\ \% / 100\ \% = 45842{,}06 \text{ (грн)}.$$

Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$А = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} [грн]. \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{вик}$ – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,90 міс.

$$А_{обл} = \frac{20000}{2} \times \frac{1,9}{12} = 1583,33 \text{ (грн)}.$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5.

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних активів є безкоштовною, то $В_{нем.ак.} = 0$ грн.

Таблиця 4.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна периферія (Acer Aspire 5 A515-45G-R5CN Charcoal Black)	20000	2	1,90	1583,333
Офісне обладнання (меблі)	21000	4	1,90	831,250
Приміщення	950000	20	1,90	7520,833
Всього				9935,42

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (4.5)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,4 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9.

$$В_е = 0,9 \cdot 0,4 \cdot 8 \cdot 38 \cdot 6,2 = 678,528 \text{ (грн)}.$$

Інші витрати та загальновиробничі витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$І_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.6)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$$І_в = 189430{,}00 * 50\% / 100\% = 94715 \text{ (грн)}.$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.7)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$Н_{нзв} = 189430{,}00 * 150\ \% / 100\ \% = 284145 \text{ (грн)}.$$

Витрати на проведення науково-дослідної роботи. Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$В_{заг} = 189430,00+18943,00+45842,06+9935,42+678,53+94715+$$
$$+284145 = 643689,00 \text{ (грн)}.$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} \text{ (грн)}, \quad (4.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 643689,00 / 0,5 = 1287378 \text{ (грн)}.$$

4.3 Розрахунок економічної ефективності науково-технічної розробки

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та

може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

– вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

– зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

– кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

– визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

– абсолютного економічного ефекту (чистого дисконтованого доходу);

– внутрішньої економічної дохідності (внутрішньої норми дохідності);

– терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.9)$$

де $\pm\Delta Ц_о$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_о$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_о = Ц_б \pm \Delta Ц_о$;

Цб – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$;

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 55000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 2000 (грн). Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1500 шт., протягом другого року – на 1000 шт., протягом третього року на 500 шт. [51].

До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1$ = (0*2000 + (55000 + 2000)*1500)* 0,8333* 0,23) * (1 - 0,18) = 12966249,481 (грн).

$\Delta П_2$ = (0*2000 + (55000 + 2000)*(1500+1000)* 0,8333* 0,23) * (1 - 0,18) = 22396249,104 (грн).

$\Delta П_3$ = (0*2000 + (55000 + 2000)*(1500+1000+500)* 0,8333* 0,23) * (1 - 0,18) = 26875498,925 (грн).

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 62237997,51 грн.

4.4 Розрахунок ефективності вкладених інвестицій і періоду їх окупності

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ = 0,05…0,15;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

ПП = (12966249,481/(1+0,1)^1)+(22396249,104/(1+0,1)^2)+(26875498,925/ /(1+0,1)^3) = 11787499,53 + 18509296,78 + 20191960,12 = 50488756,43 (грн).

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} * \text{ЗВ}, \quad (4.11)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2...5$, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 1287378 = 2574756{,}02 \text{ (грн)}.$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід *(NPV, Net Present Value)* для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$\text{Е}_{\text{абс}} = ПП - PV, \quad (4.12)$$

$$\text{Е}_{\text{абс}} = 50488756{,}43 - 2574756{,}02 = 47914000{,}41 \text{ (грн)}.$$

Оскільки $\text{Е}_{\text{абс}} > 0$то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну

внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.13}$$

де $T_ж$ – життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[3]{(1 + 47914000{,}41/2574756{,}02 \ - 1} = 1{,}697$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{4.14}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0{,}14 + 0{,}05 = 0{,}19.$$

Так як $E_в > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \tag{4.15}$$

$$T_{ок} = 1 / 1,697 = 0,59 \text{ (року)}.$$

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 0,59 року, то фінансування даної наукової розробки є доцільним.

4.5 Висновки

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 1287378 гривень.

Було спрогнозовано орієнтовану величину витрат по кожній зі статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки.

В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,59 року.

ВИСНОВКИ

У кваліфікаційній роботі розроблено інформаційну технологію передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання.

Виконано огляд важливості оцінювання будинків, охарактеризовано, що таке передбачення ціни будинків, розглянуто технології передбачення та технології машинного навчання.

Здійснено огляд відомих аналогів таких, як: DOM.RIA та ноутбуки у системі Kaggle з передбачення ціни на будинки. В результаті аналізу відомих аналогів можна підсумувати, що на ціноутворення будинків мають вплив наступні основні параметри, фактори та ознаки:

– місце розташування будинку (координати);
– площа будинку;
– кількість кімнат;
– кількість поверхів;
– рік побудови.

Результатом огляду та аналізу відомих аналогів у системі Kaggle, визначено, що точність аналогічних рішень, які використовують подібні регресійні методи машинного навчання, складає 0.81 та 0.82.

Здійснено вибір оптимальних інформаційних технологій та створено алгоритм інформаційної технології передбачення ціни будинків, яка пропонується. В якості мови програмування обрано мову Python, оскільки вона найбільше підходить для роботи з методами машинного навчання, в якості середовища розроблення програмного коду на Python обрано систему Kaggle IDE Notebooks.

Визначено, що задача передбачення ціни на будинки відноситься саме до виду машинного навчання з учителем (контрольоване навчання). А одним із найкращих варіантів розв’язку даної задачі є розв’язок за допомогою моделей регресії та моделей, які побудовані на основі дерев рішень.

Здійснено вибір та опис регресійних моделей машинного навчання, за допомогою яких відбувалось передбачення ціни продажу будинків, описано бібліотеки мови програмування Python, які використовуються для розв’язання поставленої задачі.

Виконано огляд та опис ознак вхідного набору даних та описано процес зчитування вхідного набору даних. Здійснено фільтрацію викидів та аномальних даних, визначено межі фільтрації та запропоновано правило фільтрування аномальних даних.

На етапі розвідувального аналізу даних побудовано діаграми, інтерактивні мапи та наведено різні візуалізації, які дозволили знайти в ознаках загальні закономірності, розподіли та аномалії.

Досліджено важливість ознак, побудовано діаграму важливості ознак за трьома моделями машинного навчання: LGBM, XGB, Linear regression. Визначено 5 головних ознак, які мають вплив на передбачення ціни продажу будинків:

– координати розташування будинку;
– площа земельної ділянки;
– площа житлового приміщення найближчих 15 сусідів;
– площа земельних ділянок найближчих 15 сусідів;
– рік побудови.

Підготовлено вхідні дані для побудови моделей, створено функцію для обчислення точності та похибок моделей.

Побудовано та налаштовано 7 моделей машинного навчання:

– Random Forest;
– XGB;
– LGBM;
– BaggingRegressor;
– ExtraTreesRegressor;
– Linear Regression;
– MLPRegressor.

Виконано передбачення ціни продажу будинків та порівняно моделі за точністю та похибками. Визначено оптимальну модель машинного навчання LGBM, точність передбачення якої на тренувальному наборі даних – 0.95, а на тестовому наборі даних – 0.876. Отже, що є більшим за 0.82, як у найкращого аналога (чим ближче метрика точності до 1, тим краще).

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 1287378 гривень. Було спрогнозовано орієнтовану величину витрат по кожній зі статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,59 року.

Таким чином, дістала подальший розвиток інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті з використанням методів машинного навчання, яка дозволяє підвищити точність такого передбачення у порівнянні з аналогами. Отже, завдання поставлене на магістерську кваліфікаційну роботу виконано в повному обсязі.

За результатами даної роботи написані тези доповіді, які були апробовані на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2022-2023 рр.) та опубліковані у матеріалах цієї конференції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Богачук А. Р. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ЦІНИ ПРОДАЖУ БУДИНКІВ У КІНГ-КАУНТІ МЕТОДАМИ МАШИННОГО НАВЧАННЯ. *Всеукраїнська науково-практична інтернет-конференція студентів аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)»*. Вінниця, 2022. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/viewFile/16875/14071

2. Як проводиться оцінка будинку? URL: https://pareto.com.ua/ua/blog/yak-provoditsya-ocinka-budinku/

3. Штучний інтелект. URL: uk.wikipedia.org/wiki/Штучний_інтелект

4. Третьяков М.Ю. Використання штучного інтелекту для прогнозування цін фондового ринку. Київ : КПІ ім. Ігоря Сікорського, 2019. URL: http://ktpe-conf.kpi.ua/v2019/paper/downloadSuppFile/19501/5874

5. Machine Learning. URL: https://ami.cdu.edu.ua/machine-learning/

6. Машинне навчання простими словами. URL: http://mmf.com.ua/ar/1739

7. Класифікація алгоритмів машинного навчання: лінійна регресія, класифікація та кластеризація. URL: bloginnovazione.it/uk/machine-learning/3716

8. DOM.RIA. URL: https://dom.ria.com/uk/

9. Оцінка квартири онлайн. URL: https://dom.ria.com/uk/kalkuljator-stoimosti/prodazha-kvartir/

10. Burhan Kiyakoglu, Predicting House Prices, 2018. URL: https://www.kaggle.com/code/burhanykiyakoglu/predicting-house-prices

11. House Sales in King County, USA, 2017. URL: https://www.kaggle.com/datasets/harlfoxem/housesalesprediction?datasetId=128&sortBy=voteCount

12. Rotem Gabay, House Price Predictions (R^2 0.82), 2018. URL: https://www.kaggle.com/code/rotemgb/house-price-predictions-r-2-0-82

13. Пономаренко В. С., Золотарьова І. О., Бутова Р. К., Плеханова Г. О. Інформаційні системи в економіці : навч. посіб. Харків : ХНЕУ, 2011. 176 с.

14. Інформатика та інформація. URL: https://stboinf.wordpress.com/2013/03/13/інформатика-та-інформація/

15. Інформаційні технології. URL: https://uk.wikipedia.org/wiki/Інформаційні_технології

16. Машинне навчання. URL: uk.wikipedia.org/wiki/Машинне_навчання

17. A. Bezerra, I. Silva, L. A. Guedes, D. Silva, G. Leitão, and K. Saito, "Extracting Value from Industrial Alarms and Events: A Data-Driven Approach Based on Exploratory Data Analysis," Sensors, 2019, no 19, issue 12, pp. 11-32.

18. Береза Є.Д. Переваги використання мови Python для глибокого машинного навчання. *Наукові праці Харківського національного університету радіоелектроніки.* Харків, 2021. URL: https://openarchive.nure.ua/bitstream/document/18002/1/RiM_2021_KITS_167-168.pdf

19. Підручник мови Python/Вступ. URL: uk.wikibooks.org/wiki/Підручник_мови_Python/Вступ

20. Що таке середовище розробки програмного забезпечення? URL: https://cutt.ly/U1JwNqH

21. How to Use Kaggle. URL: https://www.kaggle.com/docs/notebooks

22. Класичне машинне навчання: завдання класифікації, узагальнення, кластеризації даних. URL: https://evergreens.com.ua/ua/articles/classical-machine-learning.html

23. Machine Learning, ML. URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning

24. Жураковський Б. Ю., Зенів І. О. Технології інтернету речей : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 271 с.

25. Random Forest Regression in Python. URL: https://www.geeksforgeeks.org/random-forest-regression-in-python/

26. Regression Example with XGBRegressor in Python. URL: www.datatechnotes.com/2019/06/regression-example-with-xgbregressor-in.html

27. What is LightGBM, How to implement it? How to fine tune the parameters? URL: https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

28. Bagging and Random Forest Ensemble Algorithms for Machine Learning. URL: https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/

29. Difference between Random Forest and Extremely Randomized Trees. URL: https://stats.stackexchange.com/questions/175523/difference-between-random-forest-and-extremely-randomized-trees

30. LinearRegression. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

31. Deep Neural Multilayer Perceptron (MLP) with Scikit-learn. URL: https://towardsdatascience.com/deep-neural-multilayer-perceptron-mlp-with-scikit-learn-2698e77155e

32. Shallow and deep copy operations. URL: docs.python.org/3/library/copy

33. NumPy documentation. URL: https://numpy.org/doc/stable/

34. Pandas: powerful Python data analysis toolkit. URL: https://pypi.org/project/pandas/

35. Matplotlib documentation. URL: https://pypi.org/project/matplotlib/

36. Seaborn statistical data visualization. URL: https://pypi.org/project/seaborn

37. Plotly documentation. URL: https://pypi.org/project/plotly

38. Mapping Toolkits. URL: matplotlib.org/1.4.3/mpl_toolkits/index.html

39. SciPy. URL: https://pypi.org/project/scipy/

40. Generating Word Cloud in Python. URL: http://h1.https//www.geeksforgeeks.org/generating-word-cloud-python/

41. Folium. URL: https://pypi.org/project/folium/

42. Scikit-Learn. URL: https://pypi.org/project/scikit-learn/

43. Pandas_profiling. URL: https://pypi.org/project/pandas-profiling/

44. XGBoost Documentation. URL: https://xgboost.readthedocs.io/en/stable/

45. LightGBM Documentation. URL: https://lightgbm.readthedocs.io/en/v3.3.2

46. Warning control. URL: https://docs.python.org/3/library/warnings.html

47. Мокін В. Б. Основні прийоми аналізу даних ML/DL засобами та алгоритми Python. *Наукові праці Вінницького національного технічного університету*. Вінниця, 2019. URL: https://iq.vntu.edu.ua/method/getfile.php?fname=54790.pdf&x=1&card_id=1922

48. Штовба С.Д., Козачко О.М. Machine learning: стартовий курс : електронний навчальний посібник. Вінниця: ВНТУ, 2020. 81 с.

49. House Price Prediction by 7 models ML EDA and FE, 2022. URL: www.kaggle.com/boga4uk/house-price-prediction-by-7-models-ml-eda-fe

50. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт : уклад. Вінниця : ВНТУ, 2021. 42 с.

51. Number of Realtors in the USA by State, 2021. URL: https://hooquest.com/how-many-realtors/

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Мокін В. Б.

«19» _____09_____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ЦІНИ ПРОДАЖУ БУДИНКІВ У КІНГ-КАУНТІ МЕТОДАМИ МАШИННОГО НАВЧАННЯ»

08-53.МКР.001.02.000.ТЗ

Керівник: акад. НАПНУ, д.т.н.,
проф. каф. САІТ

_____ Мокін Б. І.

«_19_» _____09_____ 2022 р.

Розробив: студент гр. 2ІСТ-21м

_____ Богачук А. Р.

«_19_» _____09_____ 2022 р.

Вінниця 2022

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № 203 по ВНТУ від «14» _09___ 2022 р., та індивідуальне завдання на МКР, затверджене протоколом № 3 засідання кафедри САІТ від «14» ___09_____ 2022 р.

2. Джерела розробки:

- House Sales in King County, USA, 2017. URL: https://www.kaggle.com/datasets/harlfoxem/housesalesprediction?datasetId=128&sortBy=voteCount
- Emirhan Ergin, House Sales Visualization, 2020. URL: https://www.kaggle.com/code/darkcore/house-sales-visualization/notebook
- Штовба С.Д., Козачко О.М. Machine learning: стартовий курс : електронний навчальний посібник. Вінниця: ВНТУ, 2020. 81 с.

3. Мета і призначення роботи:

Підвищення точності передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання шляхом створення інформаційної технології передбачення цієї ціни.

4.Вихідні дані для проведення робіт:

Набір даних з платформи Kaggle, який містить інформацію по продажу будинків у Кінг-Каунті.

5.Методи дослідження:

Методи машинного навчання, регресійні моделі та моделі, які побудовані на основі дерев рішень.

6. Етапи роботи і терміни їх виконання:

1. Аналіз предметної області ..... 20.09 – 30.09
2. Вибір оптимальних інформаційних технологій..... 01.10 – 16.10
3. Розвідувальний аналіз та обробка даних ..... 17.10 – 31.10
4. Побудова моделей та передбачення ціни будинків..... 01.11 – 13.11
5. Економічна частина ..... 14.11 – 23.11
6. Оформлення пояснювальної записки. ..... 24.11 – 30.11

7. Очікувані результати та порядок реалізації:

Розроблена інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті з використанням методів машинного навчання, яка дозволяє підвищити точність такого передбачення у порівнянні з аналогами.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

Публічний захист ..... «_19_» _____12_____ 2022 р.

Початок розробки..... «_20_» _____09_____ 2022 р.

Граничні терміни виконання МКР ..... «_30_» _____11_____ 2022 р.

Розробив студент групи 2ІСТ-21м _____ Богачук А. Р.

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія передбачення ціни продажу будинків у Кінг-Каунті методами машинного навчання»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Науковий керівник: Мокін Б. І., акад. НАПНУ, д.т.н., проф. каф. САІТ

Показники звіту подібності Unicheck

Оригінальність	81,7 %
Схожість	18,3 %

Аналіз звіту подібності (відмітити потрібне):

- ☑ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Опис прийнятого рішення:

Робота допускається до захисту

Особа, відповідальна за перевірку Жуков С. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Богачук А. Р.

Керівник роботи Мокін Б. І.

Додаток В

Лістинг програми

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from plotly.offline import init_notebook_mode,iplot
from mpl_toolkits import mplot3d
from scipy import stats
from wordcloud import WordCloud
# map visualization
import folium
from folium import plugins
# preprocessing
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV,
StratifiedKFold
import pandas_profiling as pp
# models
from sklearn.linear_model import LinearRegression, SGDRegressor, RidgeCV
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.preprocessing import LabelEncoder, StandardScaler
import xgboost as xgb
import lightgbm as lgb
```

```
features = ['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront',
'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipc
ode', 'lat', 'long', 'sqft_living15', 'sqft_lot15']

valid_part = 0.3
pd.set_option('max_columns',100)
train0 = pd.read_csv('../input/housesalesprediction/kc_house_data.csv')
train0 = train0[features]
train0.head(5)

numerics = ['int8', 'int16', 'int32', 'int64', 'float16', 'float32', 'float64']
categorical_columns = []
features = train0.columns.values.tolist()
for col in features:
    if train0[col].dtype in numerics: continue
    categorical_columns.append(col)
for col in categorical_columns:
    if col in train0.columns:
        le = LabelEncoder()
        le.fit(list(train0[col].astype(str).values))
        train0[col] = le.transform(list(train0[col].astype(str).values))

str_list = [] # empty list to contain columns with strings (words)
for colname, colvalue in train0.iteritems():
    if type(colvalue[1]) == str:
         str_list.append(colname)
num_list = train0.columns.difference(str_list)
house_num = train0[num_list]
f, ax = plt.subplots(figsize=(16, 12))
plt.title('Pearson Correlation of features')
```

```
sns.heatmap(house_num.astype(float).corr(),linewidths=0.25,vmax=1.0,
square=True, cmap="cubehelix", linecolor='k', annot=True)

drop_columns = ['zipcode', 'view', 'waterfront', 'yr_renovated']
train0 = train0.drop(columns = drop_columns)

train0 = train0[(
                (train0['price'] <= 1000000) &
                (train0['price'] > 170000) &
                (train0['bathrooms'] <= 4) &
                (train0['condition'] > 2.5) &
                (train0['grade'] != 4) &
                (train0['sqft_lot15'] > 1300) &
                (train0['sqft_lot15'] < 44000) &
                (train0['sqft_lot'] > 1500) &
                (train0['sqft_lot'] < 70000) &
                (train0['sqft_living'] > 700) &
                (train0['yr_built'] > 1925) &
                (train0['bedrooms'] > 0) &
                (train0['bedrooms'] < 7)
            )]

feature_score = pd.DataFrame(
        preprocessing.MinMaxScaler().fit_transform(feature_score),
        columns=feature_score.columns,
        index=feature_score.index
        )
feature_score['mean'] = feature_score.mean(axis=1)
feature_score.sort_values('mean', ascending=False).plot(kind='bar', figsize=(20, 10))

target_name = 'price'
```

```
train_target0 = train0[target_name]
train0 = train0.drop([target_name], axis=1)

train0, test0, train_target0, test_target0 = train_test_split(train0, train_target0, test_size=0.2, random_state=0)

scaler = StandardScaler()
train0 = pd.DataFrame(scaler.fit_transform(train0), columns = train0.columns)

acc_train_r2 = []
acc_test_r2 = []
acc_train_d = []
acc_test_d = []
acc_train_rmse = []
acc_test_rmse = []

def acc_d(y_meas, y_pred):
    return mean_absolute_error(y_meas, y_pred)*len(y_meas)/sum(abs(y_meas))
def acc_rmse(y_meas, y_pred):
    return (mean_squared_error(y_meas, y_pred))**0.5

random_forest = RandomForestRegressor()
random_forest.fit(train, target)
acc_model(0,random_forest,train,test)

xgb_clf = xgb.XGBRegressor({'objective': 'reg:squarederror'})
    parameters = {'n_estimators': [60, 100, 120, 140],
                  'learning_rate': [0.01, 0.1],
                  'max_depth': [5, 7],
                  'reg_lambda': [0.5]}
    xgb_reg = GridSearchCV(estimator=xgb_clf, param_grid=parameters, cv=5, n_jobs=-1).fit(trainb, targetb)
```

```
        print("Best score: %0.3f" % xgb_reg.best_score_)
        print("Best parameters set:", xgb_reg.best_params_)
acc_boosting_model(1,xgb_reg,trainb,testb)

params = {
        'boosting_type':'gbdt',
        'objective': 'regression',
        'num_leaves': 31,
        'learning_rate': 0.01,
        'max_depth': -1,
        'subsample': 0.8,
        'max_bin' : 5000 ,
        'bagging_freq': 20,
        'colsample_bytree': 0.6,
        'metric': 'rmse',
        'min_split_gain': 0.5,
        'min_child_samples': 10,
            }
modelL = lgb.train(params, train_set = train_set, num_boost_round=10000,
early_stopping_rounds=8000,verbose_eval=500, valid_sets=valid_set)
acc_boosting_model(2,modelL,trainb,testb,modelL.best_iteration)

bagging = BaggingRegressor()
bagging.fit(train, target)
acc_model(3,bagging,train,test)

linreg = LinearRegression()
linreg.fit(train, target)
acc_model(5,linreg,train,test)

mlp = MLPRegressor()
```

```
param_grid = {'hidden_layer_sizes': [i for i in range(2,20)],
                'activation': ['relu'],
                'solver': ['adam'],
                'learning_rate': ['constant'],
                'learning_rate_init': [0.01],
                'power_t': [0.5],
                'alpha': [0.0001],
                'max_iter': [1000],
                'early_stopping': [True],
                'warm_start': [False]}
mlp_GS = GridSearchCV(mlp, param_grid=param_grid, cv=10, verbose=True, pre_d
ispatch='2*n_jobs')
mlp_GS.fit(train, target)
acc_model(6,mlp_GS,train,test)

models = pd.DataFrame({ 'Model': ['Random Forest', 'XGB', 'LGBM', 'BaggingRegre
ssor', 'ExtraTreesRegressor', 'Linear Regression', 'MLPRegressor'],
      'r2_train': acc_train_r2,
      'r2_test': acc_test_r2,
      'd_train': acc_train_d,
      'd_test': acc_test_d,
      'rmse_train': acc_train_rmse,
      'rmse_test': acc_test_rmse
})

lgb_predict = modelL.predict(test0)
lgb_predict[:3]
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ЦІНИ ПРОДАЖУ БУДИНКІВ У КІНГ-КАУНТІ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Виконав: студент гр. 2ІСТ-21м

_____ Богачук А. Р.

«01» _____12_____ 2022 р.

Керівник: акад. НАПНУ, д.т.н.,
проф. каф. САІТ

_____ Мокін Б. І.

«02» _____12_____ 2022 р.

Нормоконтроль: к.т.н., доцент

_____ Жуков С. О.

«02» _____12_____ 2022 р.

Рисунок Г.1 – Блок-схема алгоритму інформаційної технології передбачення ціни на будинки

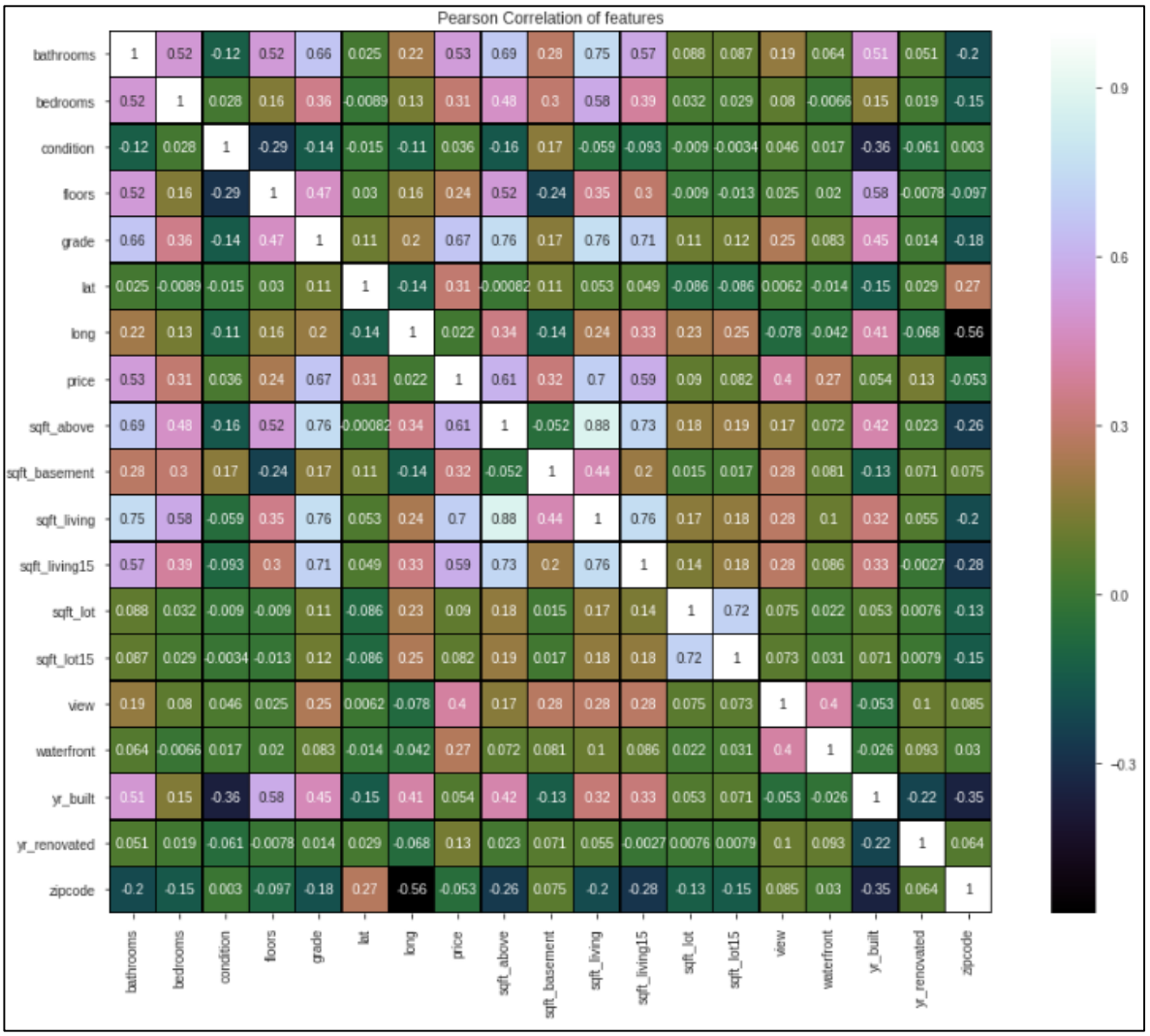

Рисунок Г.2 – Теплова кореляційна матриця ознак

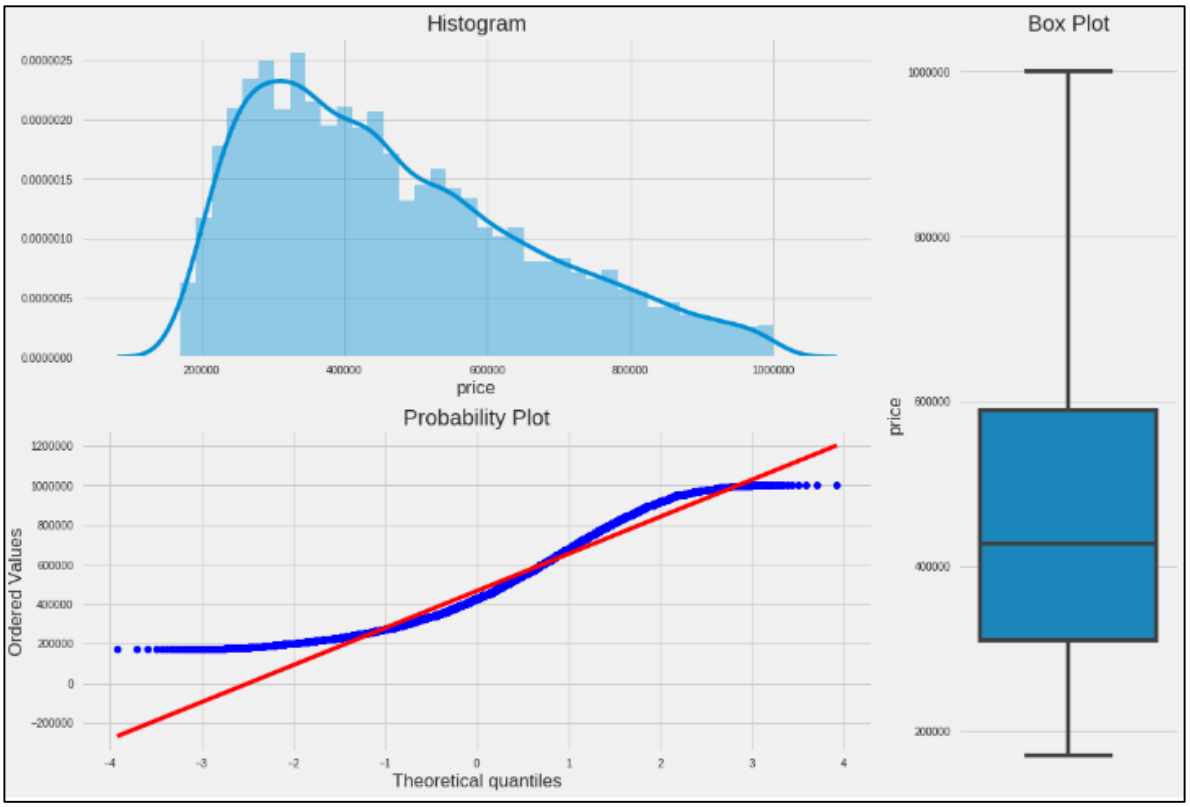

Рисунок Г.3 – Перевірка на аномальні дані методами Matplotlib, Pandas та Seaborn

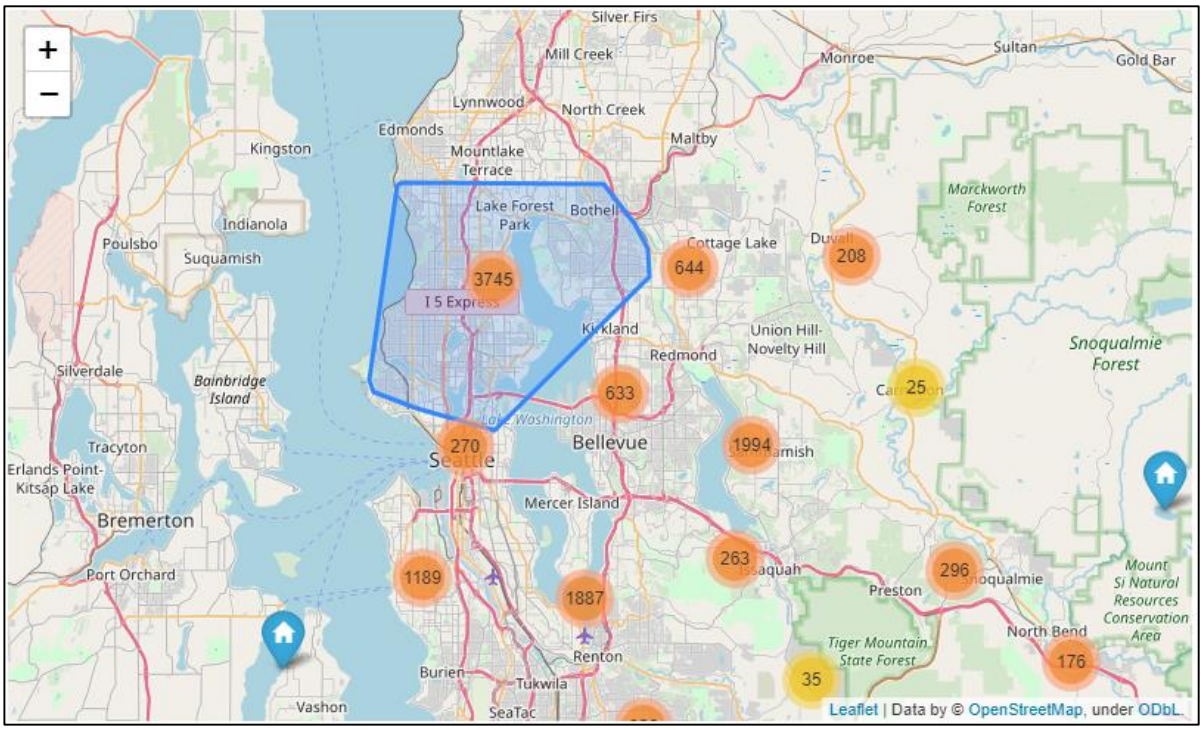

Рисунок Г.4 – Інтерактивна мапа розташування будинків

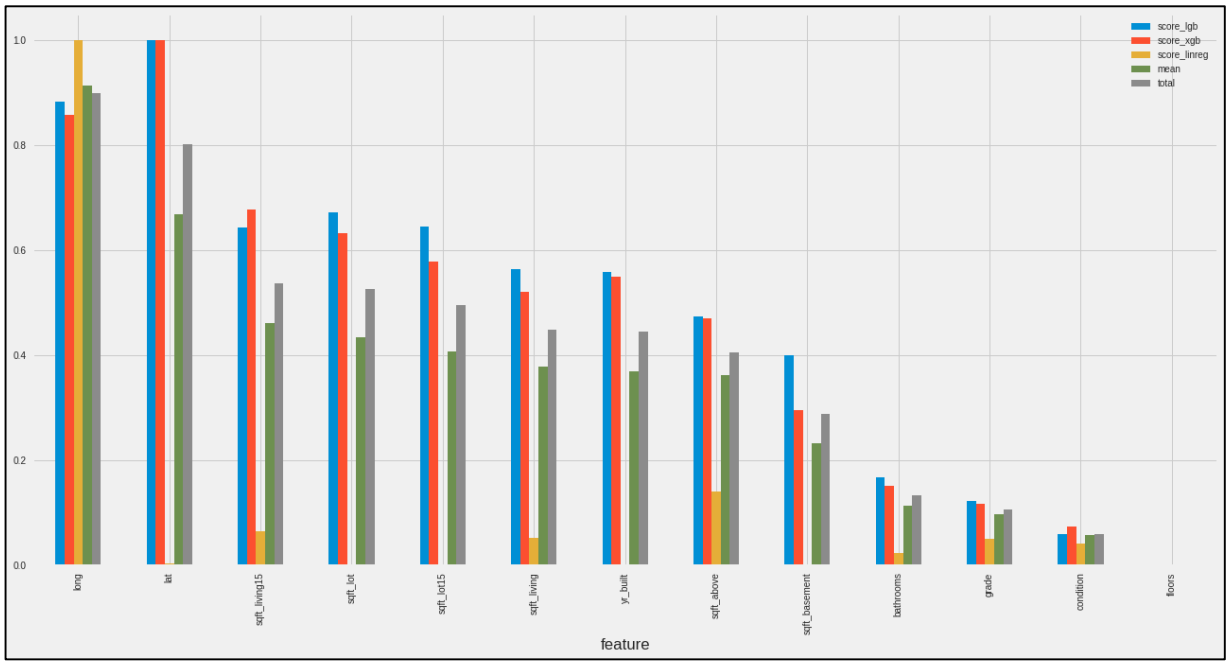

Рисунок Г.5 – Діаграма важливості ознак

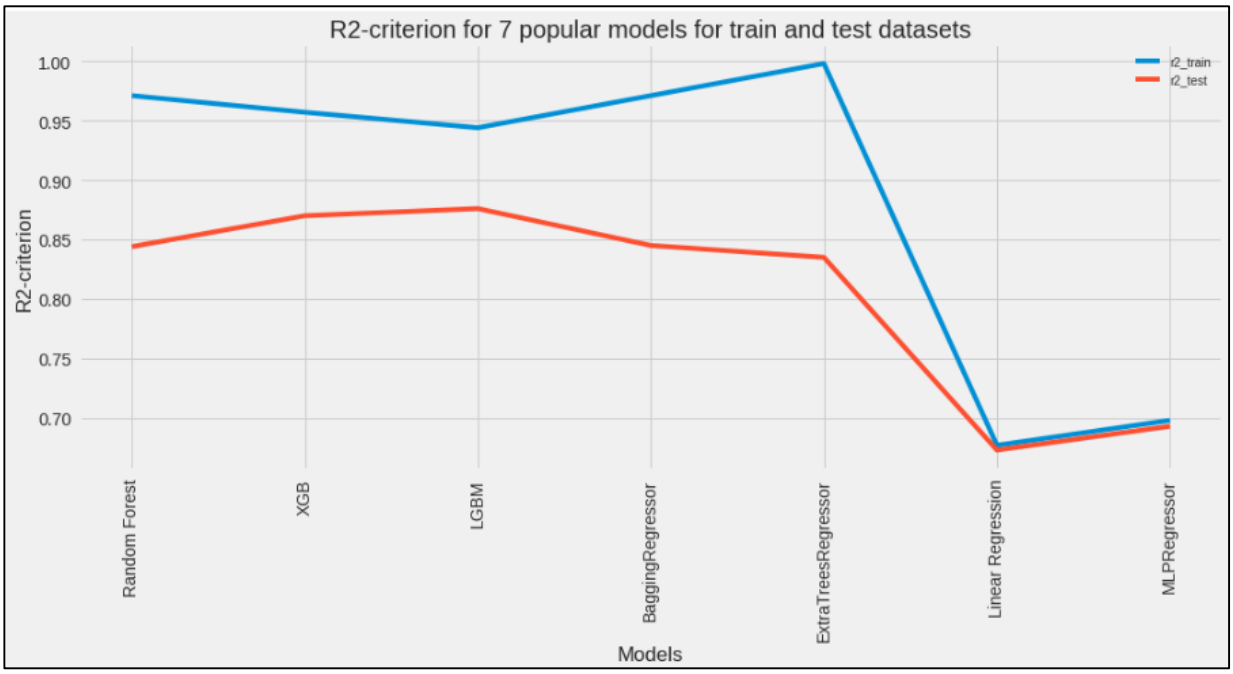

Рисунок Г.6 – Графік точності моделей