Міністерство освіти і науки України

Ніжинський державний університет імені Миколи Гоголя

Навчально-науковий інститут природничо-математичних,

медико- біологічних наук та інформаційних технологій

Кафедра інформаційних технологій, фізико-математичних та

економічних наук

Освітня програма: Комп’ютерні науки

Спеціальність:122 Компʼютерні науки

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття освітнього ступеня *магістр*

Розвиток машинного навчання та

його застосування в різних практичних задачах і наукових дослідженнях

студента **Труби Олександра Миколайовича**

**Науковий керівник:**

Казачков Іван Васильович,

доктор технічних наук, професор

**Рецензент:**

**Допущено до захисту:**  2023 р. Завідувач кафедри

проф. Казачков І.В.

Ніжин − 2023

АНОТАЦІЯ

Метою кваліфікаційної роботи є аналіз розвитку машинного навчання та демонстрація його потенціалу в різних областях застосування. Об'єктом дослідження є алгоритми машинного навчання, їх структура та області застосування. Предмет роботи – застосування машинного навчання в практичних задачах.

Магістерська робота складається зі вступу, основної частини, яка включає два розділи, висновків, списку використаних джерел.

У першому розділі проводиться детальний аналіз історії та основних етапів розвитку машинного навчання, його основних концепцій та методологій. У другому розділі розглянутий практичний приклад застосування машинного навчання, для якого підготовлено декілька моделей. Після цього проведений порівняльний аналіз результатів кожної моделі з метою вибору найкращої.

За результатами роботи було зроблено висновки та сформульовано рекомендації, спрямовані на підвищення ефективності застосування машинного навчання в практичних задачах. Особлива увага приділяється вибору алгоритмів та підходів, які оптимізують обробку і аналіз даних, що є актуальним для різних задач.

**Ключові слова:** машинне навчання, випадковий ліс, аналіз даних.

ABSTRACT

The purpose of the qualification work is to analyze the development of machine learning and demonstrate its potential in various fields of application. The object of study is machine learning algorithms, their structure and areas of application. The subject of the work is application of machine learning in practical tasks.

The master's thesis consists of an introduction, the main part, which includes two sections, conclusions, a list of sources used.

The first section provides a detailed analysis of the history and main stages of machine learning development, its basic concepts and methodologies. The second section discusses a practical example of machine learning application, for which several models have been prepared. After that, a comparative analysis of the results of each model was carried out in order to select the best one.

Based on the results, conclusions are made and recommendations are formulated aimed at improving the efficiency of machine learning in practical tasks. Particular attention is paid to the choice of algorithms and approaches that optimize data processing and analysis, which is relevant for different tasks.

**Keywords:** machine learning, random forest, data analysis.

# ЗМІСТ

[ЗМІСТ 3](#_7dyhep5dls5h)

[ВСТУП 4](#_mh5mb1qwpogz)

[РОЗДІЛ 1 6](#_73d57ttkq3h6)

[1.1 Розвиток машинного навчання 6](#_m2ldsobvqb0b)

[1.2 Складові машинного навчання 10](#_e0bdhs6ysvbl)

[1.3 Класифікація методів машинного навчання 12](#_r7fh9bg7ii74)

[1.3.1 Класичне навчання 14](#_ogbh6gg04moq)

[1.3.2 Навчання з підкріпленням 21](#_ionij4ustkhv)

[1.3.3 Ансамблі 22](#_hlfqautpgy1d)

[1.3.4 Нейромережі і глибоке навчання 23](#_w4khuttbevnf)

[1.4 Застосування в практичних задачах та наукових дослідженнях 24](#_ql040yac4m7q)

[1.4.1 Приклади застосування класичного навчання 24](#_yjt5dkjxz7xq)

[1.4.2 Приклади застосування навчання з підкріпленням 26](#_ekji09j1ww9k)

[1.4.3 Приклади застосування ансамблів 26](#_6yx89xh6zxr7)

[1.4.4 Приклади застосування нейромереж і глибокого навчання 27](#_tk0fy9ie8j57)

[1.5 Висновки розділу 28](#_ne2pb0yh9qmf)

[РОЗДІЛ 2 29](#_lfhgrvv6884q)

[2.1 Аналіз даних 29](#_8vtjbrhzv2ch)

[2.2 Підготовка даних 35](#_tau0aalr87lw)

[2.3 Навчання моделей 36](#_wrh0g2psid90)

[2.4 Висновки розділу 40](#_vt5eo3cv0slr)

[ВИСНОВКИ 41](#_jxsb2f2ky3mi)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 42](#_el7ois5zs95a)

# ВСТУП

**Актуальність теми.** У сучасному світі даних та технологій, машинне навчання відіграє ключову роль у багатьох галузях, стаючи одним з найбільш перспективних і динамічно розвиваючихся напрямків у сфері штучного інтелекту. З його допомогою можна аналізувати великі обсяги даних, розпізнавати складні закономірності та робити точні прогнози, що є важливим для прийняття рішень у бізнесі, науці, медицині та багатьох інших сферах.

Особливо актуальним є застосування для аналізу та прогнозування цін на житло, що становить велику частку економіки багатьох країн. Цей сектор характеризується великою кількістю факторів, які впливають на ціноутворення, включаючи економічні показники, географічне розташування, інфраструктуру, історичні дані та багато інших. Традиційні методи аналізу часто не в змозі ефективно обробляти всю цю інформацію, тоді як машинне навчання може забезпечити глибший та більш точний аналіз.

Застосування машинного навчання у цій сфері може допомогти інвесторам, розвідникам, компаніям з нерухомості та звичайним покупцям краще розуміти ринкові тенденції, робити обгрунтовані прогнози та приймати виважені рішення. Це може також сприяти створенню більш прозорих та ефективних ринків нерухомості.

У цьому контексті розвиток та застосування машинного навчання для аналізу та прогнозування цін на житло відкриває значні можливості для дослідження, інновацій та реалізації практичних рішень, що робить цю тему вкрай актуальною для вивчення та розвитку.

**Об’єкт дослідження –** алгоритми машинного навчання, їх структура та області застосування.

**Предмет дослідження** – ефективність моделей машинного навчання в практичних задачах.

**Метою** кваліфікаційної роботи є дослідження ефективність різних моделей машинного навчання в практичних задач та стратегій обробки даних.

Відповідно до вказаного об’єкту, предмету і для досягнення поставленої мети визначені такі **завдання** роботи:

* проаналізувати технічну, спеціальну літературу відповідно до теми кваліфікаційної роботи;
* підготувати тестові дані;
* підготувати моделі на основі зібраних даних;
* проаналізувати результати для підготовлених моделей;
* зробити висновки щодо ефективності підготовлених моделей машинного навчання.

**Методи дослідження.** Аналітичні методи оброблення інформації. Методи статистичного аналізу. Методи машинного навчання.

**Наукова новизна отриманих результатів** полягає в тому, що дана робота дозволяє оцінити ефективність різних моделей машинного навчання для прогнозування цін на житло, з акцентом на аналізі впливу різноманітних економічних та соціальних факторів для актуальних даних. Існуючі подібні дослідження зазвичай обмежуються побудовою моделей на історичних даних, що не завжди ефективно.

Робота має **практичне значення,** оскільки вони допомагають покращити точність та ефективність прогнозування цін, що є важливим для інвесторів, розробників та кінцевих споживачів.

**Апробація результатів дослідження.**

Опубліковано статтю “Застосування машинного навчання для аналізу та прогнозування цін на житло” у електронному збірнику наукових праць молодих учених факультету математики, природничих наук та технологій Центральноукраїнського державного університету імені Володимира Винниченка “Наукові записки молодих учених” [1].

# РОЗДІЛ 1

**ОСНОВНІ КОНЦЕПЦІЇ МАШИННОГО НАВЧАННЯ**

## Розвиток машинного навчання

Машинне навчання – це підгалузь штучного інтелекту, яка зосереджується на розробці систем, здатних автоматично вчитися та покращувати свою ефективність із часом, використовуючи дані. Цей процес не потребує явного програмування для виконання конкретної задачі. Натомість, моделі машинного навчання розвиваються на основі аналізу великих обсягів даних, ідентифікуючи закономірності та тренди, які можуть бути використані для прийняття рішень або прогнозів.

У традиційному програмуванні під час виконання завдання комп’ютер слідує заздалегідь визначеним інструкціям. Однак у машинному навчанні системі дається набір прикладів, за допомогою яких їй необхідно з’ясувати, як розв’язати проблему. Наприклад, щоб комп’ютер розпізнавав кішок на зображеннях, можна не надавати йому конкретних інструкцій про те, який вигляд має тварина. Замість цього потрібно показати алгоритму кілька відповідних фотографій, за якими він виявить загальні закономірності та особливості, що визначають кота. Згодом, під час обробки більшої кількості картинок модель поступово тренується, щоб точно розпізнавати таких тварин навіть на невідомих зображеннях [2].

Основні типи машинного навчання включають навчання під наглядом, навчання без нагляду та підсилювальне навчання.

Навчання з учителем передбачає, що система тренується знаходити закономірності на власному прикладі. Як правило, весь цей процес контролює інженер. Під час навчання алгоритму надають величезні масиви розмічених даних, наприклад, зображення рукописних цифр з анотаціями, що вказують на їхній відповідний номер. За наявності достатньої кількості прикладів система навчиться розпізнавати кластери пікселів і форм, пов’язаних із кожним об’єктом, і в результаті зможе з високою точністю ідентифікувати їх на фотографіях. Але, як вже було згадано, такий метод вимагає використання великої кількості тренувальної інформації. Іноді алгоритму необхідно “згодувати” мільйони прикладів, щоб домогтися високої точності його роботи. Також величезний масив навчальних даних необхідно розмітити, а це досить трудомісткий процес. Часто для цього залучають працівників краудсорсингових платформ на кшталт Amazon Mechanical Turk [2].

Навчання без вчителя спрямоване на тренування алгоритму виявляти закономірності у вхідній інформації. Система тренується виявляти подібності в даних і розділяти їх на категорії. Прикладом таких систем є модель онлайн-майданчика Airbnb, яка групує доступні для оренди будинки за районами. Так само агрегатор новин Google News створює щоденні добірки статей на схожі теми. Алгоритми навчання без вчителя не призначені для виділення конкретних типів даних. Вони просто шукають інформацію, яку можна згрупувати за схожістю, або виявляють аномалії [2].

Навчання з підкріпленням передбачає, що ШІ-агенти тренуються ухвалювати рішення, взаємодіючи з певним середовищем. За кожні вжиті дії їх винагороджують або карають балами. Мета агента – максимізувати загальну кількість очок. Щоб зрозуміти суть навчання з підкріпленням, необхідно подумати про те, як людина вперше грає в комп’ютерну гру без знань правил і управління. Вона може виявитися абсолютним новачком, але дивлячись на взаємозв’язок між натисканнями кнопок, тим, що відбувається на екрані, і рахунком, продуктивність геймера буде збільшуватися з кожним рівнем. Прикладом системи навчання з підкріпленням є алгоритм DQN, розроблений ШІ-лабораторією DeepMind. Ця модель перемогла людей у широкому спектрі класичних відеоігор. DQN отримує пікселі з кожної гри й аналізує різну інформацію про її стан, наприклад, відстань між об’єктами. Потім алгоритм визначає, як управління впливає на те, що відбувається на екрані, і співвідноситься з набраною ним кількістю очок. У процесі багатьох циклів гри система будує модель того, як певні дії та за яких обставин дають змогу максимізувати рахунок і отримати винагороду [2].

Машинне навчання як галузь бере свій початок у ранніх дослідженнях штучного інтелекту. У 1950 році Алан Тюрінг опублікував свою знамениту статтю “Computing Machinery and Intelligence”, започатковуючи дебати про можливість машин мислити. Тюрінг запропонував тест, який пізніше назвала в честь нього, який був створений задля визначення здатності машини проявляти інтелектуально обумовлену поведінку, яка схожа до поведінки людини і таку, що неможливо відрізнити від поведінки людини. Стандартна інтерпретація цього тесту звучить наступним чином: Суддя взаємодіє з одним комп'ютером і однією людиною. На підставі відповідей на питання суддя повинен визначити, з ким він розмовляє: з людиною чи з комп'ютерною програмою. Завдання комп'ютерної програми – ввести суддю в оману, змусивши зробити неправильний вибір” [3]. У цей час відбувається активна робота над першими програмами, здатними вчитися. Однією з перших була програма, яку створив Артур Семюел у 1959 році для гри в шашки. Ця програма могла покращувати свої навички, аналізуючи попередні ігри. Програма аналізувала партії і вивчала ходи, що складають виграшну стратегію. Вона використовувала ці дані в наступних матчах. У 1962 році майстер із гри в шашки Роберт Нілі воював із програмою Семюеля. Самонавчальна система виграла тільки першу партію, але стала монументальним технологічним досягненням того часу. Семюел називав машинне навчання “галуззю, яка дає комп’ютерам здатність навчатися без наявності в них точної програми”.

У 1960-ті і 1970-ті роки машинне навчання продовжило свій розвиток. Важливим внеском було створення перцептрону Френк Розенблат, який є однією з ранніх форм нейронних мереж. Цей простий алгоритм був здатен навчатися і виконувати класифікаційні завдання, хоча й мав свої обмеження. Елементарний перцептрон складається з елементів трьох типів: S-елементів, A-елементів та одного R-елементу [4].

S-елементи – це шар сенсорів, або рецепторів. У фізичному втіленні вони відповідають, наприклад, світлочутливим клітинам сітківки ока або фоторезисторам матриці камери. Кожен рецептор може перебувати в одному з двох станів – спокою або збудження, і лише в останньому випадку він передає одиничний сигнал до наступний шару, асоціативним елементам [3].

A-елементи називаються асоціативними, тому що кожному такому елементові, як правило, відповідає цілий набір (асоціація) S-елементів. A-елемент активізується, щойно кількість сигналів від S-елементів на його вході перевищує певну задану величину . Сигнали від збуджених A-елементів, своєю чергою, передаються до суматора R, причому сигнал від i-го асоціативного елемента передається з коефіцієнтом . Цей коефіцієнт називається вагою A-R зв'язку. Так само як і A-елементи, R-елемент підраховує суму значень вхідних сигналів, помножених на ваги (лінійну форму). R-елемент, а разом з ним і елементарний перцептрон, видає 1, якщо лінійна форма перевищує поріг , інакше на виході буде -1. Схематично це відображено на рисунку 1.1 [3].

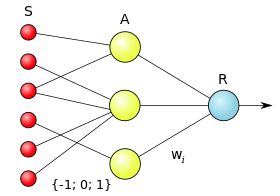


Рисунок 1.1 – Логічна схема елементарного перцептрону

Навчання елементарного перцептрона полягає у зміні вагових коефіцієнтів зв'язків A-R. Ваги зв'язків S-A (які можуть приймати значення (-1; 0; 1)) і значення порогів A-елементів вибираються випадковим чином на самому початку і потім не змінюються. Після навчання перцептрон готовий працювати в режимі розпізнавання або узагальнення. У цьому режимі перцептрону пред'являються раніше невідомі йому об'єкти, й він повинен встановити, до якого класу вони належать. Робота перцептрона полягає в наступному: при пред'явленні об'єкта, збуджені A-елементи передають сигнал R-елементу, що дорівнює сумі відповідних коефіцієнтів . Якщо ця сума позитивна, то ухвалюється рішення, що даний об'єкт належить до першого класу, а якщо вона негативна – то до другого [3].

Кінець 1970-х і початок 1980-х років були позначені скороченням інтересу і фінансування в області штучного інтелекту та машинного навчання. Це було викликано високими очікуваннями, які не були виправдані, та обмеженнями технологій того часу. Наступні роки позначили відродження інтересу до машинного навчання. Це було частково викликано розвитком нових алгоритмів, таких як зворотне поширення помилок, що дозволило навчати більш складні нейронні мережі. У цей період також відбулись значні успіхи в області обробки природної мови та автоматичного розпізнавання мовлення, що дало поштовх для подальших досліджень.

У 2000-ті роки, зі зростанням обчислювальної потужності та доступності великих даних, машинне навчання вступило у нову еру. Глибоке навчання, підгалузь машинного навчання, стало основним напрямом досліджень. Проривні алгоритми глибокого навчання, такі як згорткові нейронні мережі, змінили підходи в зоровому розпізнаванні. Це також був період, коли машинне навчання стало важливим інструментом у багатьох галузях, від медицини до фінансів. На сьогодні ця галузь продовжує розвиватися з неймовірною швидкістю. Глибоке навчання стає все більш складним і потужним, дозволяючи створювати системи, здатні до складного аналізу та обробки даних. Інновації в області обробки природної мови відкривають нові можливості для автоматизації та взаємодії між машинами та людьми. Разом з цим зростає увага до етичних аспектів машинного навчання, особливо в контексті приватності, упереджень та контролю.

## Складові машинного навчання

Мета машинного навчання полягає в аналізі та прогнозуванні результатів на основі обробки вхідних даних. Різноманітність та обсяг цих даних сприяє підвищенню точності прогнозів, оскільки це дає можливість алгоритмам виявляти складніші закономірності. Три основні компоненти, які необхідні для ефективного машинного навчання: дані(data), ознаки(features) та алгоритми(algorithms) [5].

Дані – якість та кількість даних має вирішальне значення. Наприклад, для розпізнавання спаму потрібні зразки спам-повідомлень, для прогнозування ринкових тенденцій — історичні дані про ціни акцій тощо. Ефективність моделі машинного навчання залежить від обсягу та якості зібраних даних. Також застосовуються методи автоматичного збору даних та креативні підходи, як-от використання інструментів, подібних до ReCaptcha, для генерації великих обсягів анотованих даних [5].

Ознаки – це ключові атрибути даних, які модель використовує для навчання. Вони можуть включати числові, категорійні та текстові дані, як-от пробіг автомобіля, стать користувача чи частота вживання певних слів. Вибір та обробка ознак вимагають уважного аналізу, оскільки надмірність ознак може призвести до зниження ефективності та швидкості роботи моделі [5].

Вибір алгоритму залежить від специфіки задачі, типу даних та бажаної точності моделі. Різні алгоритми можуть вирішувати однакові завдання з різною ефективністю. Важливо розуміти, що навіть найсучасніші алгоритми не можуть ефективно працювати, якщо вони тренуються на некоректних або недостатніх даних.

Таким чином, синергія між даними, ознаками та алгоритмами є ключовою для успішного застосування машинного навчання(рисунок 1.1).



Рисунок 1.2 – Компоненти машинного навчання

Якщо розглядати машинне навчання, як самостійний компонент, то гарним прикладом буде рисунок 1.2, який візуалізує відношення між різними поняттями в області штучного інтелекту та машинного навчання. Найбільше коло позначає штучний інтелект, як загальну область, що включає всі типи технологій, які здатні імітувати або відтворювати людські пізнавальні процеси.

Всередині цього великого кола знаходиться менше коло, що позначає машинне навчання, яке є підмножиною штучного інтелекту. Сюди входять алгоритми та статистичні моделі, що дозволяють комп'ютерам виконувати завдання без явно програмованих інструкцій, навчаючись з даних. Далі міститься ще менше коло з підписом нейромережі, яке є ще вужчою областю машинного навчання. Нейромережі намагаються імітувати роботу людського мозку для поліпшення власної здатності до навчання та робити більш точніші прогнози.

В останньому, найменшому колі, виокремлено глибоке навчання, що є підмножиною нейромереж, зосереджене на складних нейронних мережах з багатьма шарами, що дозволяють вирішувати ще більш складні завдання.

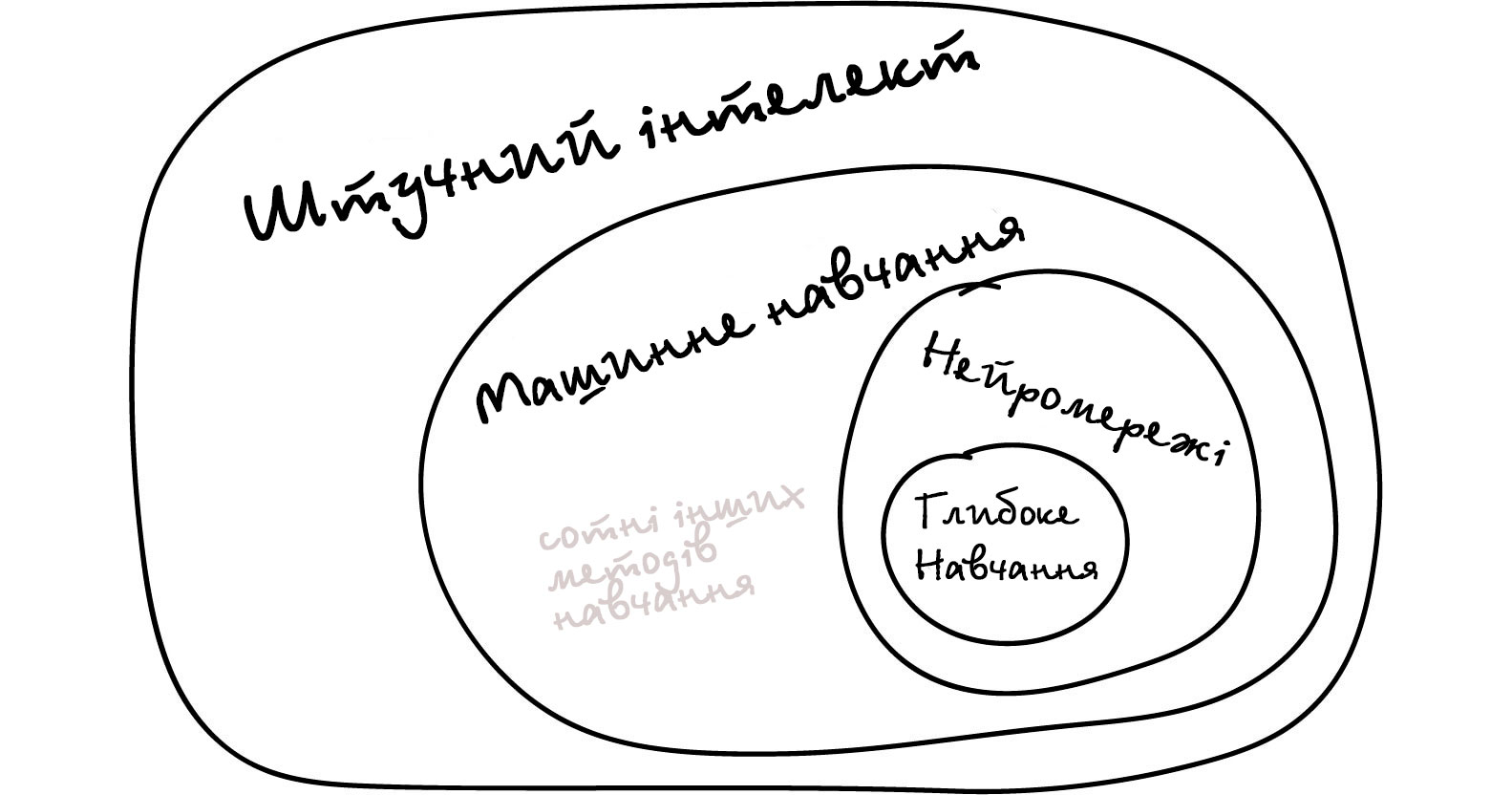


Рисунок 1.3 – Структура штучного інтелекту

## Класифікація методів машинного навчання

Методи машинного навчання можна класифікувати за різними особливостями: від того, чи потрібні вхідні дані з мітками до способів навчання моделей на основі взаємодії з довкіллям. Один з таких розділів зображено на рисунку 1.4.

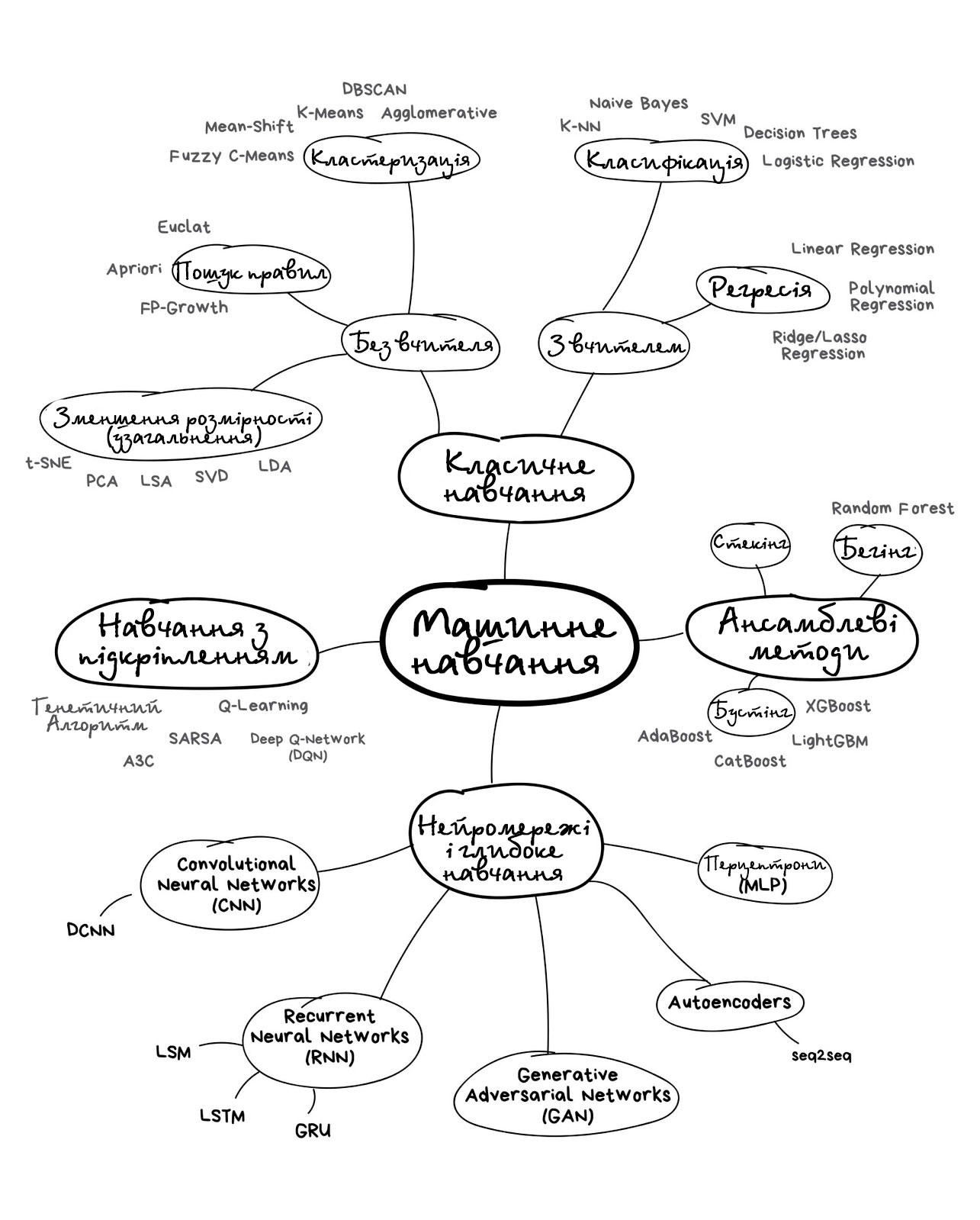


Рисунок 1.4 – Ментальна карта машинного навчання

На сьогодні в машинному навчанні виділяють чотири основні напрями, які представленні на рисунку 1.5.

# 

Рисунок 1.5 – Основні типи машинного навчання

Розглянемо детальніше кожен з наведених типів машинного навчання.

### 1.3.1 Класичне навчання

Підхід класичного навчання, основою якого є класичні статичні алгоритми. Воно розвʼязує проблеми, що стосуються прийняття рішень на основі даних. Початкові алгоритми з'явилися ще у 1950-х роках. Дані алгоритми вирішували формальні завдання, шукали закономірності у числах, а також оцінювали відстані між точками у просторі та визначали напрями певних відрізків. Ознайомитися з класифікацією підвидів машинного навчання можна на рисунку 1.6.

Класичні алгоритми є не лише дуже популярними, але й простими. На сьогодні у машинному навчанні їх можна прирівняти до основ арифметики, які є дуже поширеними та використовуються інженерами на щоденній основі. Часто ці алгоритми використовуються для завдань прогнозування та класифікації.

​​

Рисунок 1.6 – Ієрархія підвидів машинного навчання

Класичне навчання поділяється на навчання з вчителем та навчання без вчителя. У свою чергу, підвидами навчання з вчителем є класифікація та регресія. Навчання без вчителя поділяється на кластеризацію, зменшення розмірності та асоціацію.

Навчання з вчителем (Supervised Learning) є одним із основних підходів в машинному навчанні, де модель навчається на основі набору даних, які складаються з вхідних ознак та відповідних міток чи вихідних значень. Цей процес називається навчанням через те, що модель намагається вивчити способи взаємозв'язку між вхідними ознаками та цільовими відповідями, щоб у майбутньому розробити прогнози чи класифікацію для нових вхідних даних. Модель отримує навчальний набір даних, який складається з прикладів вхідних даних та відповідних правильних відповідей чи міток. Після того обирається певний тип моделі чи алгоритм, який буде навчатися на цих даних. Це може бути лінійна регресія, дерева рішень, наївний Баєсів класифікатор, метод опорних векторів (SVM), нейронні мережі та інші. У процесі модель починає навчатися на вхідних даних, використовуючи алгоритм оптимізації для підгонки параметрів чи ваг моделі з метою мінімізації помилок передбачень на навчальних даних. Після навчання модель перевіряється на тестовому наборі даних, щоб оцінити її загальну ефективність та уникнути перенавчання, коли модель вивчає шум чи деталі, які не мають загальної відповідності. Після успішного навчання та перевірки модель може бути використана для прогнозування чи класифікації нових, раніше невідомих даних.

Навчання з вчителем використовується для задач, де потрібно передбачати числові значення (регресія) або класифікувати дані у певні категорії (класифікація). Цей метод широко застосовується у сферах фінансів, медицини, рекомендаційних систем, обробки природної мови, комп'ютерного зору та багатьох інших сферах.

Регресія в машинному навчанні використовується для передбачення числових значень на основі вхідних даних. Головна мета полягає в тому, щоб встановити відношення між залежною змінною, яку потрібно передбачити, та незалежними змінними – вхідними ознаками чи характеристиками. Відомо такі методи регресії: лінійна регресія, дерева рішень, метод опорних векторів. Прогнозування значень: Регресія спрямована на передбачення числових значень, таких як ціни, кількість, час, температура тощо, на основі залежності між вхідними ознаками та значенням, яке ми намагаємося передбачити. Регресія допомагає знайти залежність між змінними. Мета – знайти математичну функцію чи модель, яка найкраще відображає зв'язок між вхідними даними та цільовою змінною. Ця функція може бути лінійною, нелінійною, поліноміальною тощо. Також регресія оцінює точність прогнозування – після навчання моделі важливо оцінити її точність на нових даних. Для цього застосовуються метрики, які дозволяють визначити, наскільки добре модель передбачає значення.

Регресія має різні методи побудови моделей, включаючи лінійну регресію, дерева рішень, метод опорних векторів (SVM), нейронні мережі та інші. Вибір конкретного методу залежить від природи даних та самої задачі. Після успішної побудови моделі вона може бути використана для передбачення чи моделювання значень для нових даних, які не були використані при навчанні.

Регресія є потужним інструментом для розв'язання проблем передбачення числових значень у різних областях, таких як економіка, фінанси, медицина, наука про дані, аналіз ринків та інші. Вона допомагає розуміти зв'язок між різними факторами та передбачати майбутні події на основі цих відносин.

Класифікація в машинному навчанні використовується для призначення категорії або класу новим даним на основі вхідних ознак. Головна мета полягає в тому, щоб побудувати модель, яка може відрізняти одні категорії від інших. Методами класифікації є логістична регресія, наївний Байєсів класифікатор, методи ближніх сусідів (k-Nearest Neighbors - k-NN). Класифікація робить можливим призначення класу або категорії для нових даних на основі інформації, яку модель вивчила під час навчання. Модель вчиться розпізнавати певні шаблони, властивості чи характеристики, що відрізняють один клас від іншого на основі вхідних ознак. Класифікація може бути бінарною, коли розділяються лише два класи, або багатокласовою, коли існує більше двох категорій для призначення.

Важливо оцінити точність та ефективність моделі на нових даних за допомогою метрик класифікації, таких як точність, відгук, F-міра, матриця помилок та інші, для оцінки її працездатності. Після успішної класифікації модель може бути використана для призначення категорії для нових даних, що не були використані під час навчання.

Класифікація використовується у різних сферах, таких як медицина, фінанси, реклама, відповідь на спам електронної пошти, визначення категорій продуктів та багато інших. Цей підхід дозволяє автоматизувати процес класифікації та призначення категорій на основі аналізу даних, оскільки головна мета класифікації – побудувати модель, яка може відрізняти одні категорії від інших.

Навчання без вчителя (Unsupervised Learning) – це розділ машинного навчання, де модель працює з даними без наявності цільових міток чи відповідей. Основна ідея полягає в тому, щоб система самостійно знаходила внутрішні закономірності, шаблони чи структури у наборі даних без заздалегідь наданої інформації. На початку, як і в усіх типах машинного навчання, відбувається підготовка даних для подальшої обробки. Це може включати очищення, масштабування та візуалізацію даних для аналізу. Далі обирається відповідний алгоритм для обробки даних відповідно до конкретної задачі. Це може бути алгоритм кластеризації, пошук асоціативних правил, або методи зменшення розмірності. Дані надсилаються на вхід моделі, яка аналізує їх та намагається виявити внутрішні зв'язки, структуру чи складність. Наприклад, алгоритм кластеризації розділить дані на групи, або алгоритм зменшення розмірності може зменшити кількість ознак, зберігаючи при цьому важливу інформацію.

Після застосування моделі важливо оцінити її ефективність та корисність. Це може включати використання метрик, які допомагають визначити, наскільки добре модель адаптується до даних. Отримані результати можуть бути використані для подальшого аналізу даних, створення гіпотез або виявлення корисної інформації.

Навчання без вчителя застосовується у багатьох областях, включаючи кластерний аналіз, пошук асоціативних правил, розпізнавання патернів, аналіз тексту, обробку зображень та інші, де важко або неможливо отримати цільові мітки для даних. Моделі, які вивчаються без вчителя, допомагають виявляти цікаві зв'язки та структури в даних, що може бути корисним для подальшого розуміння даних та прийняття рішень.

Як було описано вище, навчання без вчителя класифікується на такі підвиди: кластеризація, асоціація, зменшення розмірності.

Ідея кластеризації полягає в групуванні подібних об'єктів у кластери або групи на основі їхніх властивостей чи характеристик. Основна мета полягає в тому, щоб об'єкти всередині кластеру були схожі один на одного, а об'єкти різних кластерів були якомога більше відмінні. Алгоритми кластеризації використовують метрики схожості (наприклад, відстань між об'єктами) для визначення ступеня близькості об'єктів один до одного. Чим ближче об'єкти за значенням метрики схожості, тим більше шансів, що вони потрапляють у один кластер. Після визначення схожості, алгоритми кластеризації групують об'єкти в кластери таким чином, щоб об'єкти всередині кластеру були якомога більш схожими між собою, а різні кластери були якомога більш відмінними. На відміну від навчання з учителем, де є чіткі мітки класів, кластеризація не потребує попередньо визначених міток. Алгоритм самостійно розділяє дані на основі їхніх властивостей, а не відомої інформації про класи.

Деякі алгоритми потребують заздалегідь визначення кількості кластерів, тоді як інші можуть автоматично визначати оптимальну кількість кластерів на основі внутрішніх характеристик даних. Після кластеризації можна проводити подальший аналіз кожного кластеру, виявляти внутрішні закономірності чи залежності, або використовувати ці групи для різних завдань аналізу даних.

Алгоритми кластеризації широко використовуються в різних галузях, таких як аналіз даних, маркетинг, обробка зображень, медицина, рекомендаційні системи та інші, де важливо розподілити об'єкти на подібні групи для подальшого аналізу чи використання.

Основна ідея асоціаційного аналізу полягає у виявленні зв'язків, асоціацій або кореляцій між різними об'єктами або ознаками в наборі даних. Цей вид аналізу спрямований на пошук правил, які показують взаємозв'язки між різними елементами. Асоціаційний аналіз спочатку виявляє, як часто певні об'єкти або ознаки зустрічаються разом в наборі даних. Це може бути виконано шляхом обчислення показників, таких як підтримка, яка вказує на частоту появи певної комбінації об'єктів. Після визначення частоти взаємовходжень алгоритм створює правила асоціацій. Ці правила показують зв'язки між різними елементами чи ознаками. Наприклад, якщо покупці, які купують продукт X, також часто купують продукт Y, то може бути сформульоване правило: “Купівля Y слідує з Купівлі Х”.

Правила асоціацій мають свої метрики, такі як підтримка (support) та достовірність (confidence). Підтримка вказує на частоту появи об'єктів разом, тоді як достовірність вказує на ймовірність, що якщо відбувається одна подія, то відбудеться й інша, що з нею пов'язана. Після побудови правил асоціаційний аналіз дозволяє виявляти корисні та значущі зв'язки між об'єктами. Ці правила можуть бути використані для рекомендаційних систем, стратегій маркетингу, управління запасами, аналізу поведінки споживачів та багатьох інших областей.

Асоціаційний аналіз застосовується для виявлення цікавих зв'язків між об'єктами в даних. Цей вид аналізу допомагає виявити патерни та залежності, які можуть бути корисними для прийняття рішень, прогнозування покупок, виявлення асоціацій в медичних дослідженнях та інших сферах.

Ідея зменшення розмірності даних полягає в тому, щоб зменшити кількість ознак чи атрибутів в наборі даних, зберігаючи при цьому якнайбільше важливої інформації. Це стає актуальним у випадках, коли вихідні дані мають велику кількість ознак, що може призвести до проблеми перенавчання (overfitting), збільшення обчислювальної складності та ускладнення інтерпретації результатів.

Principal Component Analysis (PCA) – це один з найпоширеніших методів зменшення розмірності, який використовується для перетворення оригінальних ознак у новий набір ознак (принципових компонентів), де кожен принциповий компонент є лінійною комбінацією вихідних ознак. PCA зберігає найбільшу дисперсію у даних у перших нових ознаках, дозволяючи зменшити розмірність без втрати значущої інформації.

Т-distributed Stochastic Neighbor Embedding (t-SNE) – цей метод також використовується для зменшення розмірності у випадку візуалізації даних. t-SNE зберігає локальні відстані між об'єктами, створюючи нову просторову репрезентацію даних, яка зазвичай використовується для відображення високорозмірних даних у дво- або тривимірному просторі для візуалізації.

Linear Discriminant Analysis (LDA) – це метод, який враховує класи даних для знаходження лінійних комбінацій ознак, які найкраще розділяють класи між собою. Він шукає простір ознак, в якому класи мають мінімальний розсіяння всередині класу і максимальний розсіяння між класами.

Autoencoders – це тип нейронних мереж, які використовуються для навчання скраплених представлень даних. Вони вчаться стиснути вхідні дані до меншої кількості вузлів у прихованому шарі, а потім розгортати ці стиснуті представлення назад до вихідних даних.

Методи зменшення розмірності є корисними для візуалізації даних, покращення швидкості обробки та моделювання, а також для покращення загального розуміння даних шляхом зниження їхньої складності.

### 1.3.2 Навчання з підкріпленням

Навчання з підкріпленням (Reinforcement Learning) є одним з підходів у машинному навчанні, який орієнтований на навчання агента (наприклад, програмного робота чи системи) приймати послідовні рішення в певній ситуації для досягнення максимальної нагороди або досягнення певної цілі. Основна ідея полягає у тому, щоб агент навчався на основі взаємодії з оточенням і отримував відповідний відгук на свої дії.

Основними компонентами навчання з підкріпленням є: агент, середовище, дії, нагороди та стратегія. Агент – це сутність, яка взаємодіє з оточенням і вчиться виконувати дії для досягнення мети. Агент може бути програмним роботом, навченою моделлю чи іншим елементом, який приймає рішення. Середовище – це контекст, у якому діє агент. Це може бути віртуальне середовище у комп'ютерних іграх, реальна фізична система, фінансові ринки або будь-яка інша система, з якою агент взаємодіє. Дії – це можливі дії, які агент може виконати у даному середовищі. Вони можуть бути обмеженими або безліччю можливостей, які агент може приймати. Нагороди – це сигнали, які агент отримує після кожної дії у середовищі. Нагороди вказують агентові, які дії є корисними чи не корисними для досягнення мети. Стратегія – це набір правил або алгоритмів, за якими агент вибирає дії у певних ситуаціях, щоб максимізувати отримані нагороди.

Головна ідея навчання з підкріпленням полягає в тому, що агент робить спроби багато разів взаємодіяти з середовищем, навчаючись на кожному кроці, і вдосконалюючи свою стратегію, щоб максимізувати нагороду чи досягти певної цілі. Агент навчається через випробування та помилки, випробовуючи різні дії та спостерігаючи наслідки цих дій, щоб зрозуміти, які дії є найбільш ефективними.

Цей підхід до навчання використовується у різних областях, таких як робототехніка, управління, ігрові технології, фінанси, та інші, де важливо навчити агента приймати послідовні рішення в невизначеному середовищі для досягнення певної мети.

### 1.3.3 Ансамблі

Основна ідея ансамблів (ensemble) в машинному навчанні полягає в тому, щоб об'єднати декілька моделей машинного навчання для досягнення кращої точності й універсальності передбачень, ніж може дати будь-яка окрема модель. Замість того, щоб розглядати одну модель, ансамбль використовує комбінацію моделей для отримання кращих прогнозів або класифікацій.

Ансамбль може використовувати різні типи моделей машинного навчання або один і той же тип моделі з різними параметрами. Головна мета полягає у створенні різноманітності серед моделей, щоб кожна з них доповнювала і компенсувала недоліки інших. Прогнози кожної моделі об'єднуються або агрегуються для отримання кінцевого результату. Це може бути за допомогою голосування більшості (для класифікації), середнього значення (для регресії) або інших методів об'єднання результатів моделей.

Використання ансамблю може допомогти уникнути перенавчання, оскільки різні моделі в ансамблі можуть виявити різні шаблони в даних. Це дозволяє узагальнити краще, ніж окрема модель. Ансамблі часто виявляють вищу точність прогнозування або класифікації порівняно з окремими моделями, особливо коли використовуються різні моделі, які взаємно компенсують свої недоліки.

Деякі з популярних методів ансамблів включають у себе Bagging (Bootstrap Aggregating), Random Forests, Gradient Boosting, Stacking та інші. Кожен з цих методів має свої особливості у використанні та способі об'єднання результатів моделей. Наприклад, стекінг працює так, що навчаємо кілька різних алгоритмів і передаємо їх результати на вхід останньому, який приймає остаточне рішення (рисунок 1.7).

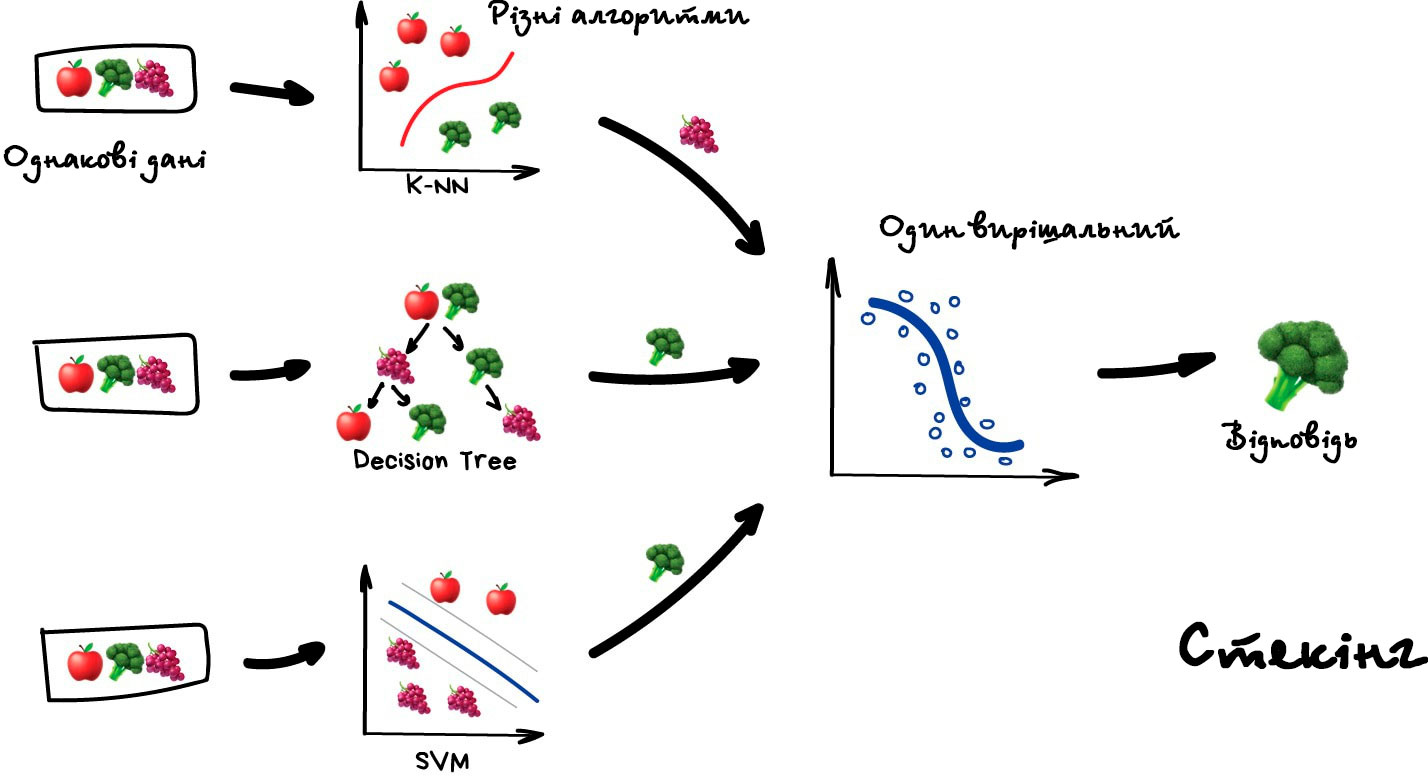


Рисунок 1.7 – Принцип робота способу стекінг

Узагальнюючи, ансамблі використовуються для покращення стабільності та точності моделей машинного навчання за рахунок об'єднання декількох моделей для отримання більш надійних та точних прогнозів чи класифікацій.

### 1.3.4 Нейромережі і глибоке навчання

Основна ідея нейромереж (neural networks) та глибокого навчання (deep learning) полягає в створенні та тренуванні моделей, які імітують роботу людського мозку. Нейромережі складаються з великої кількості з'єднаних штучних нейронів, які працюють разом для вирішення завдань машинного навчання. Кожен нейрон отримує вхідні дані, обробляє їх за допомогою ваг, здійснює певні обчислення та видає вихід. Ваги налаштовуються під час тренування, змінюючися так, щоб модель навчалася виконувати певні завдання.

Глибоке навчання використовує глибокі, зазвичай багатошарові нейромережі з багатьма шарами нейронів. Це дозволяє моделі автоматично вивчати високорівневі функції або ознаки з вхідних даних. Наприклад, у глибокому навчанні можуть використовуватися такі архітектури, як згорткові нейронні мережі (CNN) для обробки зображень або рекурентні нейронні мережі (RNN) для роботи з послідовними даними. Однією з основних переваг глибокого навчання є його здатність автоматично вивчати корисні ознаки з вхідних даних. Моделі можуть виявляти абстрактні шаблони із великих об'ємів даних, що робить їх дуже ефективними у багатьох завданнях, включаючи класифікацію, прогнозування, виявлення об'єктів у зображеннях, мовний переклад та багато іншого. Глибокі нейромережі часто тренуються за допомогою алгоритму зворотнього поширення помилки (backpropagation), де модель навчається шляхом коригування ваги нейронів, щоб мінімізувати помилку між прогнозованими та фактичними результатами.

Глибоке навчання широко застосовується в обробці зображень, обробці природної мови, аналізі даних, рекомендаціях, робототехніці, медицині, фінансах та багатьох інших областях, де потрібне складне вивчення високорівневих залежностей у великих обсягах даних.

Глибоке навчання продовжує розвиватися та знаходити нові застосування завдяки своїм потужним можливостям у розв'язанні складних завдань обробки і аналізу даних.

## Застосування в практичних задачах та наукових дослідженнях

### 1.4.1 Приклади застосування класичного навчання

Класичне навчання може використовуватися у різних видах маркетингу, офлайн і онлайн, для передбачення поведінки користувачів і надання рекомендацій, базуючись на вподобаннях користувача та його інтересах. Наприклад, коли ми вводимо у пошуковій мережі “купити сонцезахисні окуляри”, системи машинного навчання можуть пропонувати нам рекламу схожих окулярів у соціальних мережах чи на інших сайтах. Те ж саме можна спостерігати, коли ви шукаєте житло за певними критеріями: системи можуть надавати рекламу відповідних варіантів на будь-яких веб-ресурсах. Ці системи навчені автоматично підбирати варіанти, які можуть зацікавити користувача. Ці приклади ілюструють результати класичного машинного навчання.

Сьогодні велика частка мережі інтернет базується на класичних алгоритмах. Наприклад, блок “Рекомендовані статті”, що можна часто зустріти на сайтах новин або у блогах, дуже часто реалізовано саме за допомогою одного з класичних алгоритмів. Крім того, функціонал банків, що блокує гроші користувачів після першої покупки за кордоном, також у більшості випадків реалізовується на базі класичних алгоритмів машинного навчання.

Великі компанії все частіше віддають перевагу вирішенню проблем за допомогою нейронних мереж. Алгоритми класичного навчання дозволяють збільшити точність у розрахунку багатьох видів даних. Збільшивши точність навіть на два відсотки, машинне навчання дозволяє збільшити прибутки компанії на мільярди грошових одиниць прибутку. Проте для інших секторів варто бути обережнішими. Коли задачі бізнесу класичними методами розвʼязати не вдається, то краще не використовувати дані алгоритми взагалі. Прикладами, коли від класичних методів машинного навчання краще відмовитися, можуть бути випадки, описані далі.

Задачі бізнесу є надто складними для класичних методів навчання – у разі, коли задача вимагає розпізнавання дуже складних або неочевидних закономірностей у даних, класичні методи можуть бути неефективними. В таких випадках для виявлення складних залежностей краще використовувати глибоке навчання або нейронні мережі.

Об'єми даних великі – класичні методи можуть стати неефективними при обробці великих об'ємів даних. Вони можуть бути повільними та вимагати великих обчислювальних ресурсів порівняно з більш сучасними та оптимізованими алгоритмами.

Робота з низькою розмірністю даних – якщо набір даних має низьку розмірність або обмежену кількість ознак, застосування складних моделей машинного навчання може бути зайвим. У таких випадках простіші методи можуть працювати краще та швидше.

Вимоги до інтерпретованості даних – якщо потрібно мати зрозумілі результати та такі, які легко пояснити, класичні методи можуть бути більш прийнятними, оскільки нейронні мережі часто вважаються “чорними ящиками” і можуть бути інтерпретовані у меншій мірі.

Наявність простіших рішень – у випадку, коли для вирішення проблеми вже існують ефективні та прості класичні методи, застосування більш складних моделей може бути зайвим та невиправданим.

### 1.4.2 Приклади застосування навчання з підкріпленням

Навчання з підкріпленням (Reinforcement Learning, RL) використовується в різних областях, де системі потрібно приймати послідовні рішення для досягнення певної мети. RL широко використовується в ігровій технології для навчання комп'ютерних програм досягати високого рівня майстерності у відеоіграх. Наприклад, AlphaGo використовував RL для вивчення стратегій гри в Го та здолання світових чемпіонів.

Використання RL в робототехніці допомагає роботам навчитися виконувати складні завдання, такі як навчання руху, навігація у просторі, оптимізація виконання завдань. У фінансовій сфері RL може бути застосований для розробки стратегій торгівлі на фондових біржах, управління портфелем, виявлення фінансових аномалій тощо. RL може бути використане для оптимізації управління ресурсами, такими як енергія, води, транспортні системи. Наприклад, управління розподілом енергії в енергетичних мережах.

У медицині RL може використовуватися для оптимізації лікування пацієнтів, розробки нових методів діагностики, планування та оптимізації лікувальних курсів. Виробництво, виробничі процеси та автоматизовані системи управління також використовують RL для оптимізації рішень.

Усі ці приклади показують гнучкість та потенціал RL у вирішенні завдань, де важливо вивчити оптимальні стратегії прийняття рішень в змінному середовищі для досягнення певної мети.

### 1.4.3 Приклади застосування ансамблів

Ансамблі використовуються в різних галузях для поліпшення результатів прогнозування або класифікації. Вони використовуються для класифікації зображень – наприклад, у задачах розпізнавання образів, де потрібно визначити, що зображено на фотографії, ансамбль дерев приймання рішень може вирішити це завдання, об'єднуючи результати багатьох дерев.

В фінансовій сфері застосовують ансамблеві методи для прогнозування цін акцій, виявлення тенденцій у фондовому ринку та оцінки ризиків управління портфелем. Ансамблі можуть допомогти зменшити вплив шуму та покращити точність прогнозування.

Алгоритми boosting, такі як Gradient Boosting, можуть використовуватися для аналізу медичних даних та виявлення паттернів у діагностиці захворювань або визначення ефективності лікування. У задачах обробки природної мови, таких як машинний переклад або розпізнавання мови, ансамблі можуть комбінувати різні моделі для покращення точності та здатності до адаптації до різноманітних мовних структур та стилів.

У Data Science, метод стекінгу (stacking) використовує ансамблі для комбінації результатів декількох моделей машинного навчання, створюючи новий класифікатор або регресор, що може виявитися більш точним та стійким.

Ці приклади демонструють гнучкість та ефективність ансамблевих методів у різноманітних галузях, де важливо покращити результати прогнозування чи класифікації за рахунок комбінування декількох моделей.

### 1.4.4 Приклади застосування нейромереж і глибокого навчання

Нейромережі та глибоке навчання використовуються в різних галузях з різноманітними застосуваннями через їх потужність у вирішенні складних завдань обробки даних. Зокрема вони застосовуються у обробці зображень та відео. Глибокі нейромережі, такі як Convolutional Neural Networks (CNNs), використовуються для класифікації зображень у реальному часі, розпізнавання об'єктів на фотографіях або відео, наприклад, для систем відеоспостереження. Нейромережі використовуються для аналізу зображень з медичних знімків (наприклад, рентгенівських знімків, MRI), щоб виявляти хвороби або відповідати на клінічні питання.

Глибокі нейромережі, зокрема Recurrent Neural Networks (RNNs) та Transformer models, використовуються для машинного перекладу текстів з однієї мови на іншу з високою точністю. Моделі глибокого навчання можуть генерувати текст, імітуючи стиль та лексику вихідного матеріалу. Наприклад, генерація новинних статей, музики, віршів тощо.

Reinforcement Learning та нейромережі використовуються для навчання роботів виконувати складні завдання, такі як навігація у просторі, вирішення завдань у виробництві тощо. Глибокі нейромережі використовуються для аналізу фінансових даних та прогнозування цін на акції, валюту, товари тощо. Нейромережі використовуються для аналізу генетичних даних, виявлення генетичних мутацій та прогнозування хвороб.

Ці приклади демонструють широкий спектр застосування нейромереж та глибокого навчання у різних галузях, де вони забезпечують високу точність, здатність до адаптації та здатність вирішувати складні завдання обробки даних.

## Висновки розділу

У цьому розділі було розглянуто еволюцію та ключові компоненти машинного навчання, а також класифіковано різні методи, включаючи класичне навчання, навчання з підкріпленням, ансамблі та нейромережі і глибоке навчання. Детальний аналіз кожного методу виявив їхні унікальні характеристики, переваги та обмеження. Особливо важливим є розуміння того, як ці методи можуть бути застосовані для розв'язання практичних задач та у наукових дослідженнях. Через приклади застосування різних видів машинного навчання в різноманітних сценаріях можна зрозуміти, що кожен підхід має свої унікальні сфери застосування, де він може забезпечувати найкращі результати. Це свідчить про важливість глибокого розуміння особливостей кожного методу для їх ефективного відповідно до поставленої задачі.

## РОЗДІЛ 2

**ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЦІН НА ЖИТЛО**

Застосування машинного навчання для аналізу та прогнозування цін на житло відкриває нові перспективи в сфері нерухомості, надаючи потужні інструменти для оцінки вартості об'єктів, виявлення ринкових тенденцій та вдосконалення інвестиційних стратегій. Через величезний обсяг та складність даних, що включають характеристики житла, розташування, економічні індикатори та динаміку попиту та пропозиції, традиційні методи аналізу часто виявляються недостатньо ефективними. Використовуючи машинне навчання, можливо автоматизувати процес аналізу, здійснювати більш точне прогнозування цін, а також виявляти неочевидні закономірності та зв'язки між різноманітними факторами. Цей підхід не тільки підвищує прозорість і ефективність ринку нерухомості, але й сприяє кращому розумінню впливу економічних та соціальних чинників на ціноутворення в даній сфері.

## Аналіз даних

Для побудови ефективної моделі прогнозування необхідно мати обширний обсяг даних і тому для побудови моделей було обрано місто Київ, як об'єкт дослідження, вважаючи, що тут розташовано найбільше оголошень про продаж житла. Початок процесу передбачає збирання та аналіз даних, для якої використовується сайт “Лун” [6], як джерело інформації, оскільки він одним з найпопулярніших ресурсів для продажу житла. Такий підхід дозволяє отримувати актуальні дані та коригувати модель відповідно до їхньої актуальності.

Процес був досить простим, оскільки включав використання існуючих ресурсів, які надає згадай вище сайт для отримання даних. Єдиною проблемою було те, що дані можна отримати тільки частково, так є певні обмеження на завантаження максимальної кількість оголошень. Під час даної процедури загалом вдалося отримати приблизно 25 тисяч записів. Приклад даних представлено на рисунку 2.1. Варто зазначити, що ці дані заповнюються користувача, тому можливі помилки, щоб цього уникнути можна зробити додаткову фільтрацію, щоб відкинути некоректні дані. Наприклад, всі квартири ціною більше за 1 мільйон доларів або будинку, які побудовані раніше, ніж 1920 рік. Звісно, існує ймовірність, що так можна втрати певну частину інформації, але ця частка досить мала і не зашкодить подальшому аналізу.

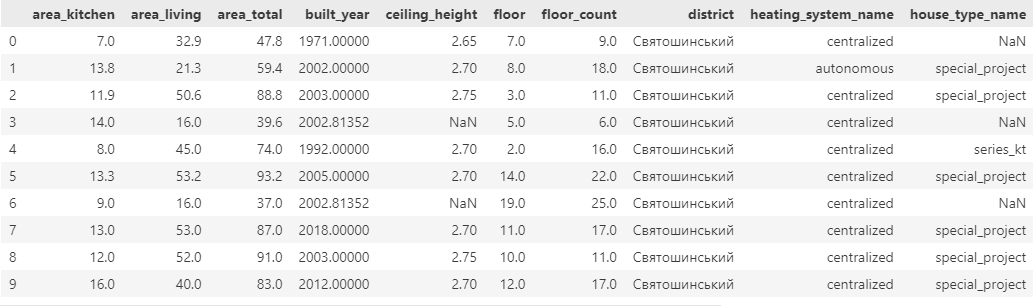


Рисунок 2.1 – Представлення даних

Для кожного запису враховується 16 параметрів, які наведені на рисунку 2.2. Це такі як: площа кухні, площа житлових приміщень, загальна площа квартири, дата побудови будинку, висота стелі, поверх, загальні кількість поверхів в будинку, район, тип опалення, тип будинку, довгота, широта, ціна, кількість кімнат, матеріал стін та чи відноситься квартира до житлового комплексу.

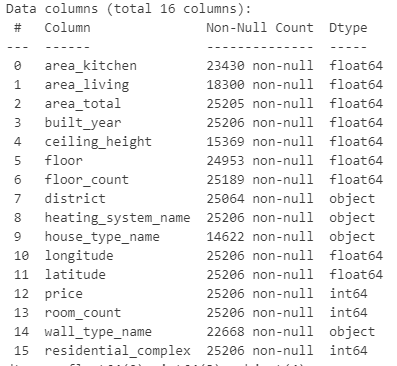


Рисунок 2.2 – Параметри вхідних даних

Як можемо бачити для деяких записів частина параметрів відсутня. Це досить очікувано та не є великою проблемою. Далі буде приведено метод, як підготувати ці дані для побудови коректної моделі. Також ми бачимо, що більшість ознак є числовими, але є такі, які визначаються обмежені певними значеннями, будемо називати їх категоріями. Давайте подивимося, скільки значень вона має характеристика район(district) (рисунок 2.3).

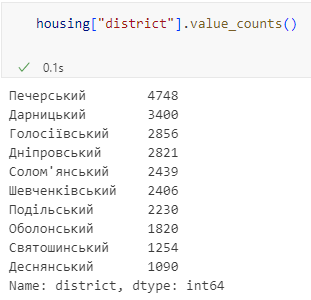


Рисунок 2.3 – Розподіл по районам

Якщо розглядати зріз по типу опалення, то, як бачимо на рисунку 2.4, більше будинків має централізоване опалення.

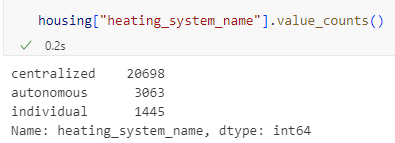


Рисунок 2.4 – Розподіл по типу опалення

Іншою цікавою характеристикою є тип будинку. При підготовці даних було з'ясовано, що вони поділяються на наступні: АППС, люкс, БПС, проект, гостинка, хрущівка, дореволюційний, совмін, серія 134, серія 87, серія 96, серія серія KП, серія КС, серія КТ, серія Т, спецпроект, сталінка. Відповідно до аналізу, представленому на рисунку 2.5, найбільше будинків за типом “спецпроект”, які будувалися в 1980-і рр.

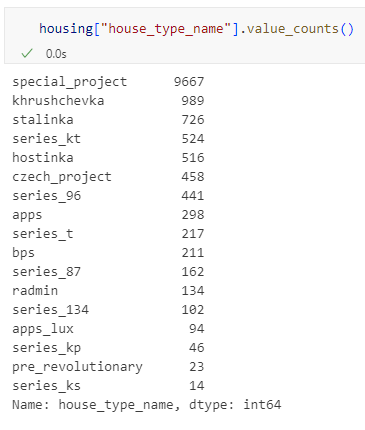


Рисунок 2.5 – Розподіл по типу будинку

Для отриманих даних проведемо детальніший аналіз. Для цього побудуємо гістограми для основих характеристик. Як можемо побачити на рисунку 2.6, що понад 8000 будинків були побудовані в 2000-х роках. Деякі характеристики виглядають прямолінійними, наприклад поверх квартири, побачимо найбільше будинків десь до 10 поверху, а потім іде вже спад. Якщо є проблема з обмеженням, ми можемо або зібрати відповідні значення для обмежень, або вилучити ці райони з наборів даних. Ми також бачимо, що не всі ознаки мають однаковий масштаб, і багато ознак мають “хвіст”, тобто простягаються на багато правіше від медіани, ніж лівіше. Це можна побачити по такому параметру як загальна площа, що може свідчити про дані з потенційною похибкою – неправильно вказана площа.

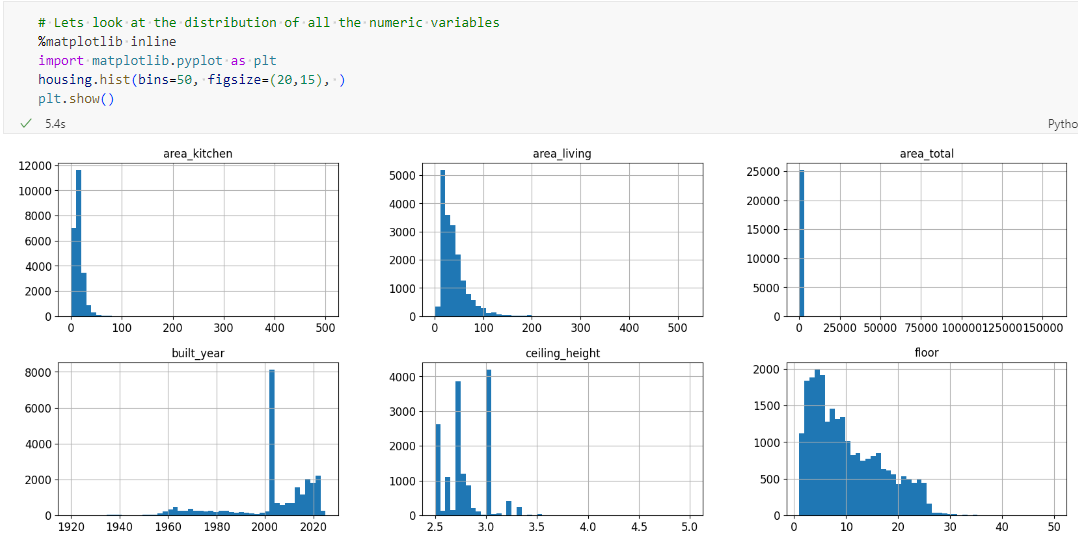


Рисунок 2.6 – Представлення даних у вигляді гістограм

Ще одним цікавим представленням є щільності забудови представлений на рисунку 2.7. На основі цього графіку, можно легко побачити районі, які мають найбільше квартир. Варто також враховувати, що це може трохи заплутати, оскільки багатоповерхові будинки матимуть більше квартир, ніж будинки з невеликою кількістю поверхів. Тому історичні райони, де будинки є досить низькими будуть відображені, як малозаселенні. Але в загальному, це гарний показник, щоб зрозуміти де проживає велика кількість людей.

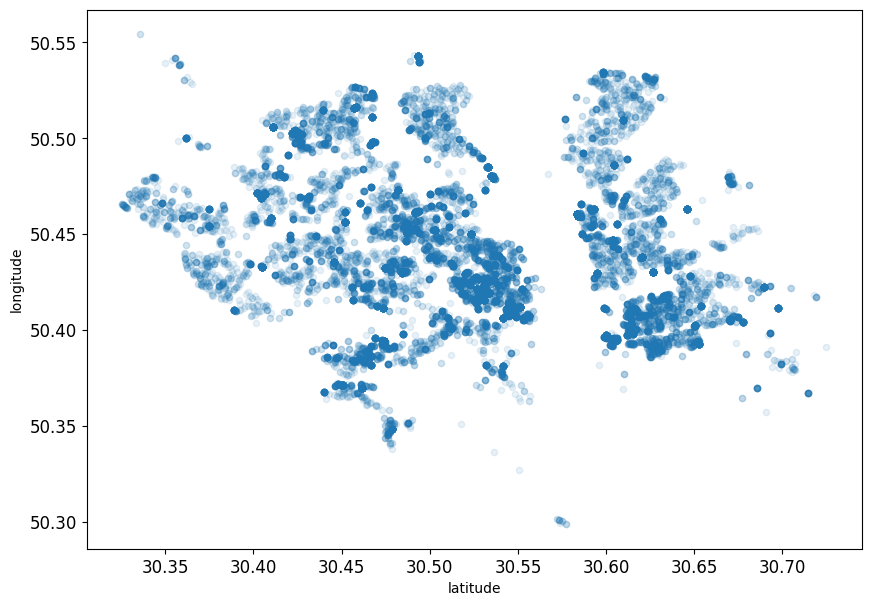
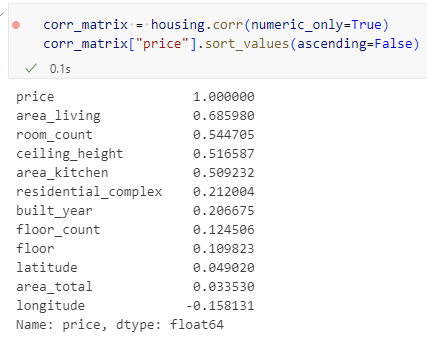


Рисунок 2.7 – Мапа щільності забудови

Наступною цікавою характеристикою для дослідження є матриця кореляцій. Матриця кореляцій – це таблиця, яка показує кореляційні коефіцієнти між змінними. Кореляційний коефіцієнт – це статистичний показник, який вимірює ступінь зв'язку між двома змінними. Значення кореляційного коефіцієнта варіюються від -1 до 1. Значення 1 означає, що між змінними існує досконала позитивна кореляція, коли одна змінна збільшується, інша теж збільшується. Значення -1 означає зворотну залежність, тобто, коли одна змінна збільшується, інша зменшується. Значення 0 означає, що між змінними немає зв'язку. У прикладі зображеному на рисунку 2.8, матриця кореляцій використовується для аналізу зв'язку між ціною житла та його характеристиками, такими як площа житлової зони, кількість кімнат, висота стелі, площа кухні та іншими. Це може бути використано, наприклад, для визначення, які фактори найбільше впливають на ціну нерухомості, що може допомогти в розробці моделі машинного навчання.



Рисунку 2.8 – Кореляція між характеристиками

На цьому етапі можна зупинитися, оскільки основний фокус в роботі буде приділятися саме на підготовці моделі машинного навчання та її оцінці. Але при потребі, цей процес можна продовжувати далі, для знаходження різних закономірностей та презентації даних у зручному форматі.

## Підготовка даних

Підготовка даних – це один з найважливіших частин завдання з машинного навчання, оскільки від того, наскільки добре ми виконаємо цей крок, залежатиме наш результат. Для цього буде використано якомога більше функцій, які допоможуть підготувати дані для подальної обробки. Перш ніж виконувати будь-яку функціональну інженерію, ми розділимо набори даних на навчальні та тестові. Зазвичай, для подібних рішень 80% даних використовуються для побудови моделі, а 20% – на тестування моделі.

Перший крок завжди полягає в тому, щоб почати з навчального набору і застосувати ті ж самі перетворення до тестових наборів. Але нам також потрібно відокремити функції від цільової змінної, оскільки в багатьох випадках вони потребують різного типу перетворень. Важливим моментом є перетворення характеристик, які мають значення категорій, у зручний для навчання, формат. Для цього ми використаємо одну техніку гарячого кодування. Це такий метод обробки даних, який часто використовується у машинному навчанні для перетворення даних у вигляді категорій на формат, який може бути краще зрозумілим для алгоритмів прогнозування. Для кожного значення категорії створюється одна бінарна характеристика. Розглянемо на прикладі характеристики район. У даному випадку нова характеристика дорівнює 1, якщо значення категорії – район для даної квартири, і 0 для всіх інших районів. У такому випадку лиш одна характеристика буде дорівнювати 1 (гарячий), тоді як інші будуть дорівнювати 0 (холодний). У результаті матимемо одну додаткову характеристику на кожне можливе значення з основної категорії.

Провівши всі маніпуляції з вхідним даними, отримаємо навчальний набір даних, який містить 20164 рядків і 47 змінних. Тепер можна перейти до найголовнішого етапу – навчання ML моделі.

## Навчання моделей

Для визначення кращої моделі було обрано декілька методів машинного навчання: лінійна регресія, дерева рішень та випадковий ліс. Після проведення досліджень було отримано результати, наведені у таблиці 1. Метод випадковий ліс показав найменше відхилення на тестових даних, тому використаємо цю модель для перевірки на тестових даних, які складають 20% від повного набору. Почнемо з базової моделі, тобто лінійної регресії. На рисунку 2.9 наведено результат моделі, оціненою за середньоквадратичне відхилення. Це популярна метрика для оцінки якості моделі. Вона вимірює середню квадратичну різницю між передбаченими значеннями, які генерує модель, та фактичними значеннями.

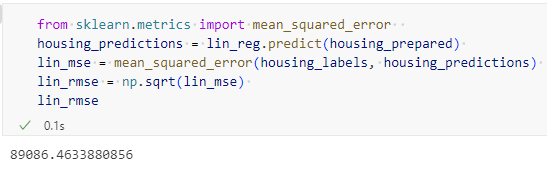


Рисунок 2.9 – Результат для лінійної регресії

Результат показує, що модель має типову похибку прогнозування у $89086, що є досить великим значенням. Це досить базова модель, то не варто зупинитися на ній. Спробуємо створити кращу модель за допомогою дерев рішень. Результат моделі наведено на рисунку 2.10.

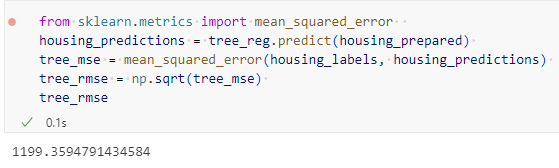


Рисунок 2.10 – Результат для дерева рішень

Щось не так, адже це виглядає занадто гарним результатом. Оскільки ми не хочемо торкатися тестового набору даних, поки не знайдемо остаточну модель, давайте використаємо техніку 10-кратної перехресної перевірки, щоб розділити навчальний набір на навчальний і перевірочний.

Перехресна перевірка – це метод оцінки якості моделей машинного навчання, який дозволяє перевірити, наскільки добре модель буде працювати на нових даних. Цей метод використовується для зменшення ризику перенавчання моделі на одному наборі даних. Для цього набір даних поділяється на кілька частин. Далі кожна частина даних по черзі використовується як тестовий набір, а решта даних – як тренувальний набір. Модель тренується на тренувальному наборі та перевіряється на тестовому наборі. Оцінки ефективності моделі на кожному тестовому наборі уважно аналізуються. Зазвичай обчислюють середнє та стандартне відхилення цих оцінок, щоб оцінити загальну якість моделі. Такий підхід дозволяє зменшити вплив випадковості, оскільки всі дані використовуються і для тренування, і для тестування, це забезпечує більш повне використання даних. І що важливо для нашого випадку, допомагає виявити моделі, які добре працюють на одному наборі даних, але погано на інших.

Після перехресної перевірки для моделі дерева рішень вдалося отримати результати зображені на рисунку 2.11. Тепер ми бачимо, що результат не дуже гарний, якщо порівнювати з лінійною регресією.

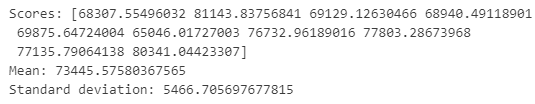


Рисунок 2.11 – Результат перевірки для дерева рішень

Для того, щоб мати повну картину, також застосуємо 10-кратну перехресну перевірку для лінійної регресії. Якщо бачимо на рисунку 2.12, модель дерева рішень дає краще рішення, хоча похибка все одно досить велика.

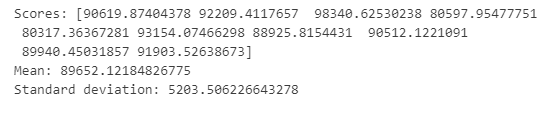


Рисунок 2.12 – Результат перевірки для лінійної регресії

Продовжуємо пошуки кращої моделі і спробуємо побудувати модель, використовуючи випадковий ліс. Випадковий ліс є потужним методом машинного навчання, який належить до категорії ансамблевих алгоритмів. Він працює, комбінуючи передбачення від декількох рішеннячих дерев, кожне з яких тренується на випадковому наборі даних, що вибирається з тестового набору. Це забезпечує підвищену точність та стійкість до помилок, а також допомагає зменшити ризик перенавчання, оскільки помилки від окремих дерев компенсуються іншими. Тому випадковий ліс дуже ефективний у вирішенні складних завдань класифікації та регресії, зокрема у випадках, коли датасет має велику кількість змінних та комплексні взаємозв'язки між ними. Крім того, це метод легко адаптується до різних типів даних та не вимагає значної попередньої обробки даних, що робить його зручним у використанні.

Побудувавши модель, отримаємо результати зображені на рисунку 2.13, що значно кращий за попередні. Помилка передбачення моделі становить лише $19829 на навчальних вибірці.

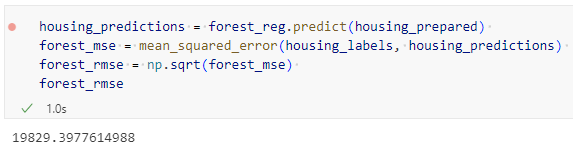


Рисунок 2.13 – Результат для випадкового лісу

Для підтвердження достовірності зробимо перевірку перевірку, як і для попередніх. Як бачимо на рисунку 2.14, це поки що одна з найкращих моделей, хоча коефіцієнт помилки досить високий у валідаційних наборах даних порівняно з навчальними наборами даних, що може свідчити про проблему надмірного підбору.

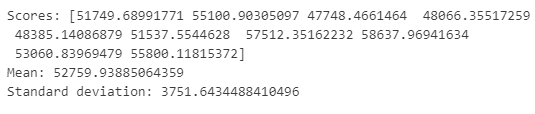


Рисунок 2.14 – Результат перевірки для випадкового лісу

Зупинимося на моделі випадкового лісу та оцінимо її на тестовому наборі. Одним із способів перевірки є використання коефіцієнту детермінація. Це коефіцієнт (R²), який вимірює, наскільки добре статистична модель прогнозує результат. Результат представлений залежною змінною моделі. Найнижче можливе значення R² дорівнює 0, а найвище – 1. Простіше кажучи, чим краще модель надає прогнози, тим ближче її R² буде до 1. Як бачимо на рисунку 2.15, для нашої моделі цей коефіцієнт складає 0,87, що є гарним показником.

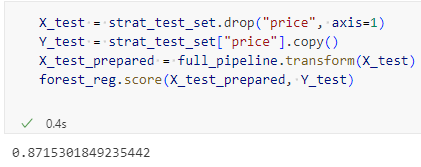


Рисунок 2.15 – Коефіцієнт детермінація

## Висновки розділу

Цей розділ демонструє гарні можливості машинного навчання у сфері аналізу та прогнозування цін на житло. Використання таких алгоритмів, як лінійна регресія, дерева рішень та випадковий ліс, дозволило нам отримати моделі, що базуються на широкому спектрі факторів, включаючи місцеположення, площу, близькість до інфраструктури та інші ключові характеристики. Оцінка моделей за допомогою метрик, таких як середньоквадратична помилка та коефіцієнт детермінації, підтвердила їхню здатність ефективно моделювати та прогнозувати ціни на житло. Особливо цінним виявився підхід перехресної перевірки, який забезпечив надійність та узагальненість отриманих результатів. Важливо відзначити, що, незважаючи на гарну точність моделей, існує потенціал для подальшого вдосконалення. Включення нових характеристики для вхідних даних, щоб точніше враховувати цільові характеристики є одним з потенційних вдосконалень. Також варто приділити увагу вхідним даним, оскільки вся інформація була взята з відкритих джерел і може мати помилки. Іншим можливим варіантом є використання більш складних моделей, які будуть враховувати значно більше параметрів, що може буде непростою задачею.

У підсумку, застосування машинного навчання у сфері аналізу та прогнозування цін на житло відкриває широкі можливості для розвитку та оптимізації ринку нерухомості, дозволяючи зацікавленим сторонам приймати обґрунтовані рішення.

# ВИСНОВКИ

Дослідження у галузі машинного навчання охоплює історію та розвиток цієї галузі, включаючи аналіз основних компонентів та різноманітних методів, з акцентом на їх особливості, переваги та області застосування. Було розглянути класифікацію методів машинного навчання та наведено ілюстративні приклади їх практичного застосування в різних галузях. У практичній частині роботи було проведена робота по збиранню та підготовці даних для створення декількох моделей машинного навчання з дослідження їх ефективність на прикладі аналізу та прогнозування цін на житло. Основними розглянутими методами були лінійна регресія, дерева рішень та випадковий ліс. Аналізуючи результати, було визначено сильні та слабкі сторони кожного методу в контексті прогнозування цін на житло та обрано найкращих з них, який надав точність 87% за вибраним критерієм оцінювання. Використання реальних даних дозволяє врахувати широкий спектр факторів, які впливають на ціну житла. Це забезпечило комплексний погляд на проблему і допомогло уникнути спотворень, які могли б виникнути при використанні обмеженого або застарілого набору даних.

Отримані результати вказують на те, що вибрані методи машинного навчання можуть бути ефективно використані для рішення задач аналізу та прогнозування в сфері нерухомості. Це дослідження підкреслює потенціал машинного навчання для удосконалення процесів прийняття рішень на ринку нерухомості та може послужити важливим кроком у напрямку створення ефективних інструментів для аналізу та прогнозу цінових тенденцій. Подібні методики можуть також бути корисними для інвесторів та ріелторів, які прагнуть оптимізувати свої інвестиційні стратегії та портфелі. Крім того, аналітики ринку нерухомості зможуть використовувати ці техніки для виявлення ключових трендів та забезпечення глибокого аналізу ринкових умов. Перспективи подібних систем здаються особливо обнадійливими з огляду на швидкий розвиток технологій та збільшення обсягів доступних даних, що дозволить забезпечити більш глибокий аналіз ринкових умов і тенденцій.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Труба О.М. Застосування машинного навчання для аналізу та прогнозування цін на житло. Наукові записки молодих учених. 2023. № 12. URL: https://phm.cuspu.edu.ua/ojs/index.php/SNYS/article/view/2037/pdf (дата звернення: 10.12.2023).
2. Що таке машинне навчання? URL: https://incrypted.com/ua/mashynne-navchannja (дата звернення: 15.11.2023).
3. Тест Тюрінга. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82\_%D0%A2%D1%8E%D1%80%D1%96%D0%BD%D0%B3%D0%B0 (дата звернення: 16.11.2023).
4. Перцептрон. URL: https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD (дата звернення: 16.11.2023).
5. Машинне навчання простими словами. URL: http://www.mmf.lnu.edu.ua/ar/1739 (дата звернення: 20.11.2023).
6. Лун. URL: https://lun.ua (дата звернення: 25.11.2023).