МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра системного аналізу та управління

АНАЛІЗ ДАНИХ ТА ЗНАНЬ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
зі спеціальності 124 Системний аналіз
(F4 Системний аналіз та наука про дані)

Дніпро
НТУ «ДП»
2025

Хабарлак К.С.

Аналіз даних та знань [Електронний ресурс] : методичні рекомендації до виконання практичних робіт для здобувачів ступеня бакалавра спеціальності 124 Системний аналіз (F4 Системний аналіз та наука про дані) / К.С. Хабарлак, Л.С. Коряшкіна, Т.А. Желдак, Т.В. Хом'як ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 62 с.

Автори:

К.С. Хабарлак, доктор філософії;

Л.С. Коряшкіна, д-р техн. наук, доц.;

Т.А. Желдак, канд. техн наук, доц.;

Т.В. Хом'як, канд. фіз.-мат. наук, доц.

Затверджено науково-методичною комісією зі спеціальності F4 Системний аналіз та наука про дані (протокол № 3 від 12.05.2025) за поданням кафедри системного аналізу та управління (протокол № 5 від 07.05.2025).

Наведено теоретичні відомості, необхідні для виконання практичних робіт згідно з робочою програмою дисципліни «Аналіз даних та знань» спеціальності 124 Системний аналіз за першим (бакалаврським) рівнем вищої освіти.

Сформульовано вимоги до оформлення звітів із практичних робіт, питання для самоконтролю та критерії оцінювання практичних робіт.

Орієнтовано на активізацію навчальної діяльності здобувачів та закріплення практичних навичок у засвоєнні дисципліни «Аналіз даних та знань».

Відповідальний за випуск завідувач кафедри системного аналізу та управління Т.А. Желдак, канд. техн. наук, доц.

Зміст

Вступ

Аналіз даних стає ключовою компетенцією для успіху бізнесу, наук і суспільства у сучасному світі, де цифрові технології та пристрої Інтернету речей генерують масивні дані. Він сприяє інноваціям, створюючи нові продукти, оптимізуючи логістику та підвищуючи ефективність. Галузь активно розвивається, збільшуючи попит на фахівців, які поєднують технічні навички з аналітичним мисленням.

У даному посібнику подано методичні рекомендації щодо виконання практичних робіт з курсу «Аналіз даних та знань». Завдання, викладені у даному посібнику, призначені для того, щоб надати студентам практичний досвід первинного аналізу, регресійного аналізу, класифікації за допомогою логістичної регресії та наївного Баєса, A/B та множинного тестування.

Завдяки вивченню зазначених тем курсу, здобувач отримає практичні навички з первинного аналізу, візуалізації, бутстрапу, довірчих інтервалів, побудови та підгонки лінійної регресії з факторними змінними, оцінки регресії з використанням перехресного контролю, A/B-тестуванню, методів множинного тестування, а також методів класифікації. Це дозволить здобувачу збирати, оброблювати та аналізувати великі масиви даних; будувати, оцінювати та застосовувати регресійні та класифікаційні моделі для передбачення на нових, невідомих вхідних даних, а також ефективно використовувати методи зменшення розмірності для оптимізації аналітичних процесів.

Мета дисципліни – сформувати у здобувачів вищої освіти: 1) практичні навички попередньої обробки, аналізу та візуалізації даних провідними методами за допомогою мови програмування Python; 2) вміння будувати моделі машинного навчання, що відповідають задачі, та навчати їх; 3) здобути навички роботи із бібліотеками Python, зокрема scikit-learn, SciPy, Pandas, NumPy, Matplotlib для машинного навчання та обробки даних. Знання та навички, отримані в курсі, будуть корисними для подальшого працевлаштування здобувача.

Завдання курсу:

- навчитися проводити первинний аналіз даних, розраховувати статистичні оцінки та візуалізувати дані за допомогою мови програмування Python;
- навчитись застосовувати машинне навчання, методи регресії та класифікації до практичних задач;
- отримати практичні навички проведення A/B тестування, множинного тестування та підтвердження статистичних гіпотез;
- опанувати роботу із мовою програмування Python для аналізу даних та із прикладними бібліотеками, scikit-learn, SciPy, Pandas, NumPy, Matplotlib.

Дисциплінарні результати навчання:

1. Вміти збирати та видобувати дані з різних джерел, структурувати та зберігати дані. Знати сучасні способи розвідувального та первинного аналізу даних, вміти застосовувати їх. Способи відбору даних.
2. Вміти розраховувати статистичні оцінки, візуалізувати розподіл даних. Оцінювати довірчий інтервал оцінки.
3. Знати сучасні методи та моделі машинного навчання, вміти навчати їх, здійснювати інтерполяцію та екстраполяцію даних.
4. Знати та вміти застосовувати основні способи візуалізації даних та їх розподілу.
5. Знати способи виявлення проблем в даних, зокрема пропущені та аномальні значення, мультиколінеарність, спотворюючі змінні. Вміти усувати їх різними способами. Вміти зменшувати просторову розмірність даних.
6. Застосовувати методи збору та аналізу даних взаємодії з користувачем (зацікавленість в веб-сторінках, кліки, конверсія). Вміти проводити A/B тест та застосовувати методи множинного тестування, здійснювати перевірку статистичної значущості їх результатів.

Практична робота №1 – Первинний аналіз даних

Мета роботи: закріпити теоретичні знання і розвинути практичні навички роботи з таблицями в Pandas, розрахунку статистик та візуалізації даних.

Практична робота присвячена розрахунку основних описових статистик при первинному аналізі даних, візуалізації даних та їх розподілу, виявленню взаємозалежностей в даних. Робота виконується із використанням мови програмування Python та бібліотек Pandas, SciPy, NumPy, Matplotlib. За бажанням студента, допускається виконання роботи мовою програмування R. Для розв'язання задачі практичної роботи за узгодженням з викладачем студент може запропонувати свій набір даних.

1.1 Теоретичні відомості

1.1.1 Типи прямокутних даних

Є два базові типи структурованих даних: числовий і категоріальний. Числові дані мають дві форми: безперервну, як, наприклад, швидкість вітру чи тривалість часу, і дискретну, як, наприклад, виникнення дії. Категоріальні дані приймають лише фіксований набір значень, як, наприклад, тип екрану телевізора (плазма, LCD, LED тощо) чи назву міста (Дніпро, Київ тощо). Двійкові дані є важливим випадком категоріальних даних. Ці дані приймають лише одне з двох значень, таких як 0/1, так/ні або істина/брехня. Ще один корисний тип категоріальних даних – порядкові дані, в яких категорії впорядковані; їх прикладом є числовий рейтинг (1, 2, 3, 4 чи 5).

Числові дані	**Категоріальні**

• Дані, що виражаються на чисельній шкалі
• **Неперервні**
  • Дані, можуть приймати будь-яке значення на інтервалі.
  • *Синоніми:* інтервал, число із плаваючою комою, чисельна величина.
• **Дискретні**
  • Дані, котрі можуть приймати лише цілочисельні значення, так як кількості.
  • *Синоніми:* ціле число, кількість, лічильна величина.
	• Дані, що можуть приймати лише конкретну множину значений, що представляють множину можливих категорій
	• *Синоніми:* перерахування, пронумеровані та номінальні дані, фактори.
	• **Двійкові**
	  • Окремий випадок категорійних даних з двома категоріями значень, наприклад, 0/1, істина/неправда.

	• *Синоніми:* дихотомічна, логічна, індикаторна, булева величина. • **Порядкові** • Категорійні дані, що мають явно вказаний порядок. • *Синонім:* впорядкований фактор.

Прямокутні дані – це загальний термін для двомірної матриці, в якій рядки позначають записи (випадки), а стовпці – ознаки (змінні). Кадр даних – це специфічний формат, властивий мовам R та Python. Вихідні дані не завжди надходять у такій формі: неструктуровані дані (наприклад, текст) необхідно обробити і привести до такого виду, щоб їх можна було подати як безліч ознак прямокутних даних. Дані в реляційних базах даних повинні бути вилучені та поміщені в одну єдину таблицю для більшості завдань з аналізу даних та моделювання.

Терміни прямокутних даних

- Кадр даних (data frame)
    - Прямокутні дані (електронна таблиця) – це базова структура даних для статистичних моделей та моделей, що автоматично навчаються.
- Ознака
    - Стовбець в таблиці зазвичай зветься ознакою.
    - *Синоніми: атрибут, вхід, провісник, предиктор, змінна.*
- Вихід
    - Багато проектів науки про дані мають на меті передбачення результату форматі так/ні. Ознаки іноді використовуються для передбачення результату експерименту або статистичного дослідження.
    - *Синоніми:* результат, залежна змінна, відгук, ціль, вихід.
- Записи
    - Рядок у таблиці зазвичай називається записом.
    - *Синоніми:* випадок, приклад, прецедент, екземпляр, спостереження, шаблон, патерн, зразок.

1.1.2 Оцінки центрального положення

Змінні з вимірюваними чи кількісними даними можуть мати тисячі чітко помітних значень. Базовий крок у розвідуванні даних полягає в отриманні «типового значення» для кожної ознаки (змінної): оцінки того, де розташована більшість даних (тобто їх центральна тенденція).

На перший погляд, завдання узагальнення даних виглядає досить тривіальною: треба просто взяти середнє арифметичне даних. Насправді незважаючи на те, що середнє обчислюється досить просто і його вигідно використовувати, воно не завжди буває найкращим підходом до обчислення центрального значення. З цієї причини у статистиці було розроблено та популяризовано кілька альтернативних оцінок середнього значення.

Середньою базовою оцінкою центрального положення є середнє значення, або середнє арифметичне:

$$\text{Середнє} = \overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1.1}$$

Різновидом середнього є середнє усічене, яке обчислюється шляхом відкидання фіксованого числа сортованих значень з кожного кінця послідовності і потім взяття середнього арифметичного значення, що залишилися. Усічене середнє усуває вплив граничних значень:

$$\text{Усічене середнє} = \overline{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p} \tag{1.2}$$

Ще один вид середнього значення – це середньозважене значення, яке обчислюється шляхом множення кожного значення даних x_i хі на свою вагу w_i поділу їх суми на суму ваг:

$$\text{Середнє зважене} = \overline{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i}^{n} w_i} \tag{1.3}$$

1.1.3 Медіана та робастні оцінки

Медіана – це число, розташоване в сортованому списку даних рівно посередині. Якщо є парне число даних, то серединним значенням є те, що не знаходиться в наборі даних фактично, а є середнім арифметичним двох значень, які поділяють сортовані дані на верхню та нижню половини.

Порівняно із середнім, у якому використовуються абсолютно всі спостереження, медіана залежить лише від значень у центрі сортованих даних. Хоча це може виглядати як недолік, оскільки середнє значення набагато чутливіше до даних, існує багато прикладів, у яких медіана є більш відповідною метрикою центрального положення. Скажімо, ми хочемо подивитись типові доходи домогосподарств в округах, розташованих на узбережжі озера Вашингтон у Сіетлі. При порівнянні округу Медіна з округом Уіндермір використання середнього значення дало б різні результати, тому що в Медіні живе Білл Гейтс. Якщо ж ми використовуватимемо медіану, то вже не буде мати значення,

наскільки багатим є Білл Гейтс, – позиція серединного спостереження залишиться тією ж.

Медіана називається робастною оцінкою центрального становища, оскільки вона не перебуває під впливом викидів (граничних випадків), які можуть спотворити результати.

Викид – це будь-яке значення, яке сильно дистанційоване від інших значень у наборі даних. Точне визначення викиду є дещо суб'єктивним, незважаючи на те, що в прикладних пакетах використовуються деякі правила.

Медіана не єдина робастна оцінка центрального становища. Насправді з метою запобігання впливу викидів широко використовується і середнє усічене. Наприклад, усічення нижніх і верхніх 10% даних (загальноприйнятий вибір) забезпечить захист від викидів у всіх, крім найменших, наборах даних. Середнє усічене може вважатися компромісом між медіаною і середнім: воно робастно до граничних значень даних, але використовує більше даних для розрахунку оцінки центрального положення.

1.1.4 Оцінки варіабельності

Центральне положення – це лише одна з розмірностей в узагальненні ознаки. Друга розмірність – варіабельність, іменована також дисперсністю, показує, чи згруповані значення даних щільно, чи вони розкидані. В основі статистики лежить варіабельність: її вимір, зменшення, розрізнення випадкової варіабельності від реальної, ідентифікація різних джерел реальної варіабельності та прийняття рішень за умов її присутності.

Так само як і у випадку центрального положення, яке можна виміряти різними способами (середнє, медіана тощо), існують різні способи виміряти варіабельність.

Терміни оцінок варіабельності

- Відхилення
    - Різниця між значенням спостережень та оцінкою центрального положення.
    - *Синоніми:* помилки, похибки, залишки.
- Дисперсія
    - Сума квадратичних відхилень від середнього, поділена на $n - 1$, де n – кількість значень даних.
    - *Синоніми:* середньоквадратичне відхилення, середньоквадратична помилка.
- Стандартне відхилення
    - Квадратний корінь з дисперсії.

    - *Синоніми:* норма l_2, *євклідова норма.*
- Середнє абсолютне відхилення
    - Середнє абсолютних значень відхилень від середнього.
    - *Синоніми:* норма l_1, мангеттенська норма
- Медіанне абсолютне відхилення від медіани
- Розмах
    - Різниця між найбільшим та найменшим значеннями в наборі даних
- Порядкові статистики
    - Метрики на основі значень даних, відсортованих від самих малих до самих крупних.
    - *Синонім:* ранги.
- Перцентиль
    - Таке значення, що P відсоток значень приймає дане значення або менше та $(100 - P)$ процент значень приймає дане значення або більше.
    - *Синонім*: квантиль.
- Міжквартильний розмах
    - Різниця між 75-м та 25-м перцентилями.
    - *Синоніми:* МКР, IQR.

$$\text{Середнє абсолютне відхилення} = \frac{\sum_{i=1}^{n}|x_i - \overline{x}|}{n}, \quad (1.4)$$

де $\overline{x}$ – середнє значення у вибірці, або вибіркове середнє.

Найвідомішими оцінками варіабельності є дисперсія та стандартне відхилення, що ґрунтуються на квадратичних відхиленнях. Дисперсія – це середнє квадратичних відхилень, стандартне відхилення – квадратний корінь з дисперсії.

$$\text{Дисперсія} = s^2 = \frac{\sum(x - \overline{x})^2}{n-1} \quad (1.5)$$

$$\text{Стандартне відхилення} = s = \sqrt{\text{дисперсія}} \quad (1.6)$$

Робастною оцінкою варіабельності є абсолютне медіанне відхилення від медіани (median absolute deviation, MAD):

$$\text{Медіанне абсолютне відхилення} = \text{медіана}(|x_1 - m|, |x_2 - m|, \ldots |x_N - m|), \quad (1.7)$$

де m– це медіана.

Оцінки на основі перцентилів. Інший підхід до оцінювання дисперсії заснований на розгляді розкиду сортованих даних. Статистичні показники з урахуванням сортованих (ранжованих) даних називаються порядковими статистиками. Елементарна міра – це розмах: різниця між найбільшим і найменшим числом. Мінімальні та максимальні значення як такі корисно знати, оскільки вони допомагають виявляти викиди, але розмах надзвичайно чутливий до викидів і не дуже корисний як загальна міра дисперсності в даних.

$$\text{Розмах} = \max_i\{x_i\} - \min_i\{x_i\} \tag{1.8}$$

Щоб уникнути чутливості до викидів, ви можете звернутися до розмаху даних після відкидання значень з кожного кінця. Ці типи оцінок формально ґрунтуються на різницях між перцентилями. У наборі даних P-й процентиль є таким значенням, що принаймні P відсотків значень приймає це значення або менше, і принаймні $(100 - P)$ відсотків значень приймає це значення або більше. Наприклад, для знаходження 80-го перцентиля треба відсортувати дані. Потім, починаючи з найменшого значення, пройти 80% вгору до найбільшого значення. Зазначимо, що медіана – це те саме, що й 50-й процентиль. Перцентиль, сутнісно, аналогічний квантилю, але квантилі індексуються частками (так, квантиль 0,8 — те саме, як і 80-й процентиль).

Загальноприйнятим способом оцінки варіабельності є різниця між 25-м та 75-м процентилями, яка називається міжквартильним розмахом (interquartile range, IQR).

Оцінки на основі перцентилів

- Розмах = $\max_i\{x_i\} - \min_i\{x_i\}$.
- Перцентиль – таке значення, що P відсоток значень приймає дане значення або менше та $(100 - P)$ процент значень приймає дане значення або більше.
- Міжквартильний розмах – різниця між 75-м та 25-м перцентилями.

1.1.5 Кореляція

Розвідувальний аналіз даних у багатьох проектах моделювання (чи то в науці про дані або в статистичному дослідженні) передбачає обстеження кореляції серед провісників і між провісниками та цільовою змінною. Прийнято говорити, що змінні X та Y (кожна з вимірюваними даними) корелюють позитивно, якщо високі значення X супроводжуються високими значеннями Y та низькі значення X супроводжуються низькими значеннями Y. Якщо високі

значення X супроводжуються низькими значеннями Y, і навпаки, то змінні корелюють негативно .

Найкорисніший стандартизований варіант – це коефіцієнт кореляції, що дає оцінку кореляції між двома змінними, які завжди знаходяться на однаковій шкалі вимірювання. Для того, щоб обчислити коефіцієнт кореляції Пірсона, ми множимо відхилення від середнього для змінної 1 на відхилення для змінної 2, а потім ділимо результат на добуток стандартних відхилень:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{(n-1)s_x s_y} \quad (1.9)$$

1.2 Приклад розв'язування задачі

1.2.1 Імпорт даних CSV та Excel у pandas

Розглянемо невеликий приклад:

people.csv

```
name,age,city
Alice,31,New York
Bob,27,Seattle
```

- Перший рядок часто розглядається як заголовок, що позначає назву кожного стовпця.
- Другий і третій рядки містять значення даних.
- Коми розділяють три колонки, а символи нового рядка завершують кожен запис.

Функція в pandas, яка виконує основну роботу, – це pandas.read_csv. Припустимо, що файл people.csv містить трирядковий приклад. У результаті буде створено DataFrame із стовпцями name, age та city і двома рядками даних.

Код

```
import pandas as pd

df = pd.read_csv('people.csv')
print(df.head())
```

Вивід

```
    name  age      city
0  Alice   31  New York
1    Bob   27   Seattle
```

Читання файлу Excel відбувається аналогічно, але використовується функція read_excel (підтримає як файли .xls так і .xlsx).

Код

```
import pandas as pd

df = pd.read_excel('people.xlsx')
print(df.head())
```

Вивід

```
    name  age      city
0  Alice   31  New York
1    Bob   27   Seattle
```

1.2.2 Розрахунок основних статистичних оцінок

Початкові дані.

Табл. 1.1. Дані США для первинного аналізу

№	Країна	Населення	Коефіцієнт народжуваності
1	Бельгія	11 556 297	1,46
2	Чехія	10 524 167	1,64
3	Німеччина	84 607 016	1,46
4	Іспанія	47 163 418	1,16
5	Франція	68 521 974	1,79
6	Румунія	19 644 312	1,71
7	Люксембург	660 809	1,31

Для обчислення середнього та медіани на Python ми можемо використовувати методи кадр даних пакета pandas. Для усіченого середнього значення потрібна функція trim_mean з бібліотеки scipy.stats:

Код для розрахунку центрального положення

```
population = pd.read_csv('population.csv')
population['Населення'].mean()
trim_mean(population['Населення'], 0.1)
population['Населення'].median()
```

Вивід

```
34668284.71
31482033.60
19644312.00
```

Зважене середнє значення доступне за допомогою пакета NumPy.

Код для розрахунку зваженого середнього	Вивід
`np.average(population['Коефіцієнт народжуваності'],weights=population['Населення'])`	1.52

Код для розрахунку стандартного відхилення	Вивід
`population['Населення'].std()`	32391892.18

Код для розрахунку перцентилів	Вивід
`population['Населення'].quantile(0.75)` `population['Населення'].quantile(0.25)`	57842696.00 11040232.00

1.2.3 Візуалізація даних

Лінійний графік:

Код

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('data.csv')

df["temp_c"].plot(
    title="Графік щоденної температури",
    ylabel="Температура",
    xlabel="Дата"
)

plt.show()
```

Результуючий графік показано на рис. 1.1.

Рис. 1.1 – Побудований лінійний графік.

Діаграма «ящик з вусами»:

Код

```
import pandas as pd
import matplotlib.pyplot as plt

state = pd.read_csv('state.csv')

ax = (state['Population']/1000000).plot.box()
ax.set_ylabel('Population (millions)')

plt.show()
```

Результат показано на рис. 1.2.

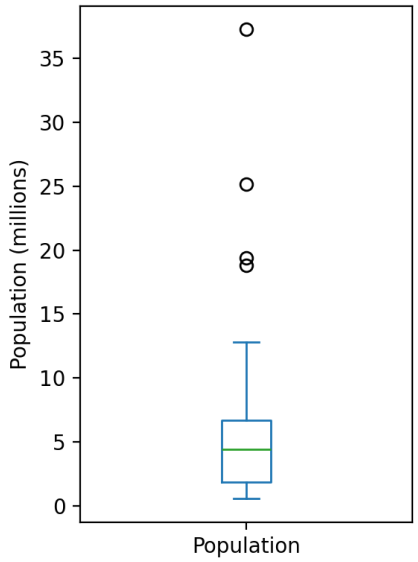

Рис. 1.2 – Коробкова діаграма, що візуалізує розподіл населення.

Гістограма:

Код

```
import pandas as pd
import matplotlib.pyplot as plt

state = pd.read_csv('state.csv')

ax = (state['Population'] / 1000000).plot.hist()
ax.set_xlabel('Population (millions)')

plt.show()
```

Отриману гістограму показано на рис. 1.3.

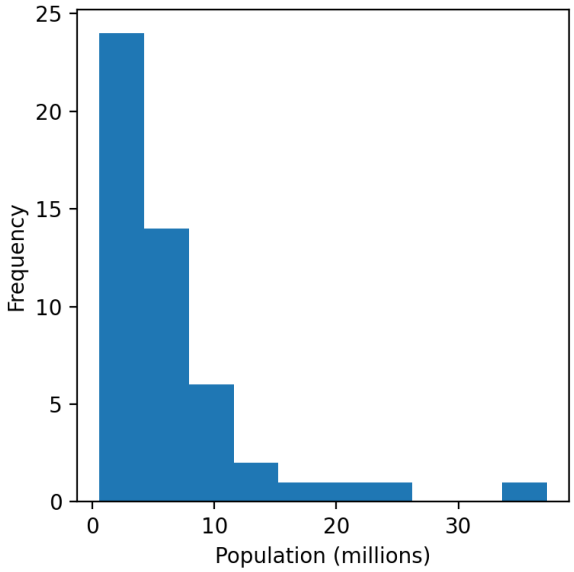

Рис. 1.3 – Гістограма розподілу населення.

Графік щільності:

Код

```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt

state = pd.read_csv('state.csv')

ax = state['Murder.Rate'].plot.hist(density=True, xlim=[0, 12],
                                    bins=range(1,12))
state['Murder.Rate'].plot.density(ax=ax)
ax.set_xlabel('Murder Rate (per 100,000)')

plt.show()
```

Результат показано на рис. 1.4.

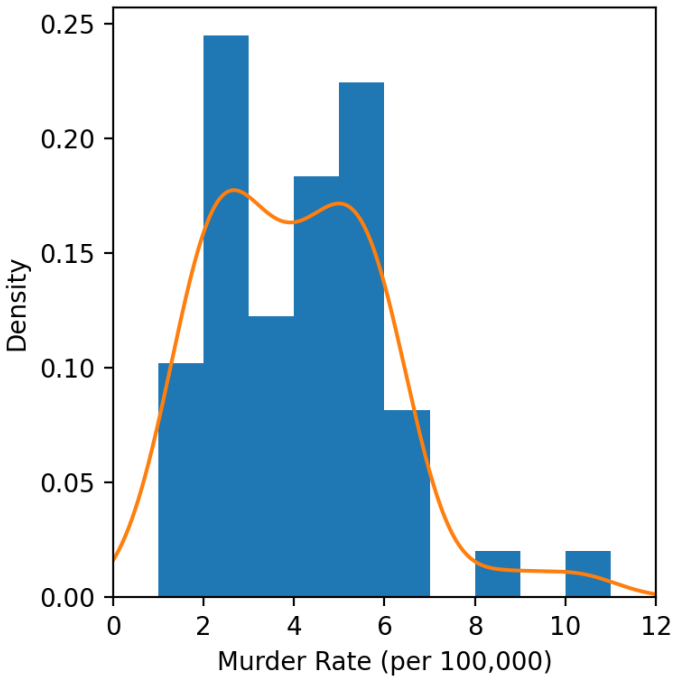

Рис. 1.4 – Графік щільності кількості вбивств.

Код

```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt

dfw = pd.read_csv('dfw_airline.csv')
print(100 * dfw / dfw.values.sum())

ax = dfw.transpose().plot.bar(legend=False)
ax.set_xlabel('Cause of delay')
ax.set_ylabel('Count')

plt.show()
```

Результат показано на рис. 1.5.

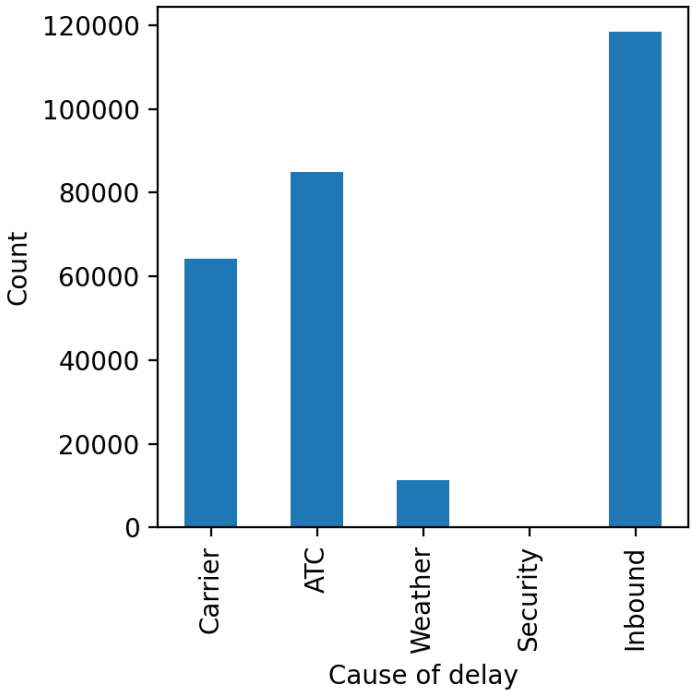

Рис. 1.5 – Стовпчикова діаграма.

Кругова діаграма:

Код

```
import pandas as pd
import matplotlib.pyplot as plt

toys = pd.read_csv('toys.csv', index_col=0)
ax = toys['sales_usd'].plot.pie()

plt.show()
```

Результат показано на рис. 1.6.

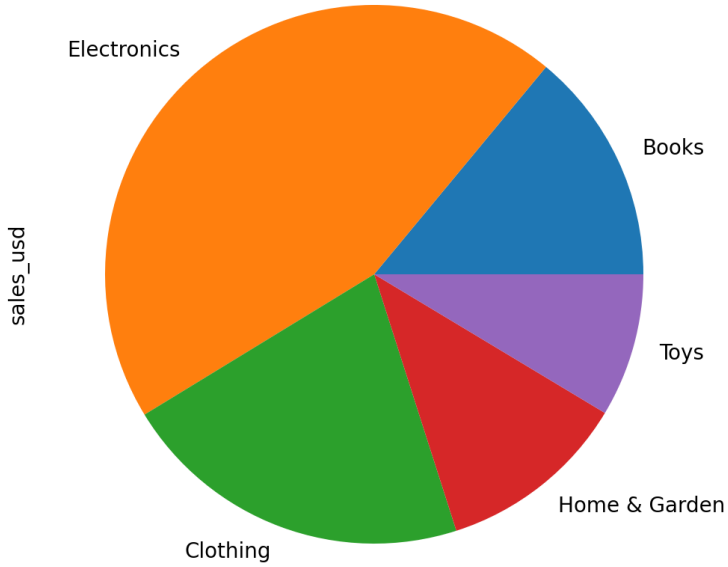

Рис. 1.6 – Кругова діаграма.

1.2.4 Кореляційна матриця

У табл. 1.2, яка називається кореляційною матрицею, показано кореляцію між щоденним доходом від акцій телекомунікаційних компаній з липня 2012 року по червень 2015 року. З таблиці видно, що Verizon (VZ) та ATT (T) мають найвищу кореляцію. Інфраструктурна компанія Level Three (LVLT) має найнижчу кореляцію. Зверніть увагу на діагональ з одиниць (кореляція акції із самою собою дорівнює 1) та надмірність інформації вище та нижче діагоналі.

Код

```
import pandas as pd
import matplotlib.pyplot as plt

telecom = pd.read_csv('telecom.csv', index_col=0)
print(telecom.head())
print(telecom.corr())
```

Табл. 1.2. Матриця кореляції.

	T	**CTL**	**FTR**	**VZ**	**LVLT**
T	1,000	0,475	0,328	0,678	0,279
CTL	0,475	1,000	0,420	0,417	0,287
FTR	0,328	0,420	1,000	0,287	0,260
VZ	0,678	0,417	0,287	1,000	0,242
LVLT	0,279	0,287	0,260	0,242	1,000

1.3 Індивідуальне завдання

1. Завантажити дані з сайту http://www.ukrstat.gov.ua/ (вкладка меню згори «Статистична інформація»). Категорію даних вибрати відповідно до варіанта:

№ варіанта	Категорія даних
1.	Населення та міграція
2.	Ринок праці
3.	Освіта
4.	Охорона здоров’я
5.	Доходи та умови життя
6.	Соціальний захист
7.	Населені пункти та житло
8.	Правосуддя та злочинність
9.	Культура
10.	Суспільна діяльність
11.	Макроекономічна статистика
12.	Національні рахунки
13.	Діяльність підприємств
14.	Послуги
15.	Внутрішня торгівля
16.	Капітальні інвестиції
17.	Основні засоби
18.	Сільське, лісове та рибне господарство
19.	Енергетика
20.	Промисловість
21.	Будівництво
22.	Транспорт
23.	Туризм
24.	Комплексна статистика
25.	Зовнішньоекономічна діяльність
26.	Ціни
27.	Наука, технології та інновації
28.	Навколишнє природне середовище
29.	Регіональна статистика
30.	Жінки та чоловіки

2. Зазначити, які існують типи структурованих даних та які представлені в ознаках таблиці.
3. Визначити, де в даних ознаки, а де – записи (див. терміни з лекції). Обрати 4 ознаки для яких розрахувати (із використанням відповідних функцій у бібліотеках), у звіті навести формули або (для деяких показників) процедуру обчислення:
    a. Середнє.
    b. Медіану.
    c. Посічене середнє:
        - для парних варіантів: із відкиданням 10% найбільших та найменших значень;
        - для непарних варіантів: із відкиданням 15% найбільших та найменших значень.
    d. Дисперсію.
    e. Стандартне відхилення.
    f. Середнє абсолютне відхилення.
    g. Мінімальне та максимальне значення.
    h. Розмах.
    i. Процентиль:
        - для парних варіантів: 99-й;
        - для непарних 95-й.
    j. Міжквартильний розмах.

Зібрати розрахунки в єдину таблицю в Word наступного вигляду:

Показник	Формула / спосіб розрахунку	Назва ознаки 1	Назва ознаки 2	Назва ознаки 3	Назва ознаки 4
Середнє					
Медіана					
Посічене середнє					
…					

4. Зробити візуалізацію даних за допомогою трьох різних типів графіків/діаграм, що є найбільш доречними для представлених даних. Обов’язково використати коробкову діаграму («ящик з вусами»), зазначити чи наявні викиди, порівняти розподіл даних декількох ознак. Для кожного з графіків навести короткий опис даних.
5. Розрахувати матрицю кореляції. Навести формулу для розрахунку кореляції Пірсона. Зазначити 2 ознаки з найбільшою кореляцією (як позитивною, так і від’ємною). Припустити, чому може існувати така кореляція.

Робота має бути представлена у вигляді звіту (файл pdf). Звіт має містити титульний аркуш, номер варіанта, розрахунки, необхідні описи та формули (відповідно до завдання вище) та вихідний код програм.

Назва файлу: Прізвище_група

Роботи, виконані не за варіантом, не приймаються.

1.4 Контрольні питання

1. Які оцінки центрального положення ви знаєте?
2. Які переваги та проблеми в простого арифметичного середнього для оцінки центрального положення?
3. Що таке викид? Яку проблему він складає під час первинного аналізу даних?
4. Що таке оцінка варіабельності?
5. Якими є основні оцінки варіабельності? Які формули для розрахунків?
6. Що таке прямокутні дані?
7. Які пакети Python передбачені для роботи з електронними таблицями, для оцінки центрального положення та варіабельності?
8. Чи є оцінки на основі перцентилів стійкими до викидів? Чому?

Практична робота №2 – Регресійний аналіз

Мета роботи: закріплення навичок збору даних, роботи із бібліотекою sklearn для побудови лінійної регресії, створення нелінійних ознак. Застосування на практиці підходу перехресної валідації.

Практична робота присвячена регресійному аналізу, підгонці регресії, створенню нелінійних ознак, способам оцінки якості. Робота виконується із використанням мови програмування Python та бібліотек Pandas, SciPy, NumPy, Matplotlib. За бажанням студента, допускається виконання роботи мовою програмування R. Для розв'язання задачі практичної роботи за узгодженням з викладачем студент може запропонувати свій набір даних.

2.1 Теоретичні відомості

2.1.1 Проста лінійна регресія

Важливим питанням аналізу даних є відповідь на запитання: чи пов'язана змінна X (або, що більш ймовірно, $X_1, \dots, X_p$) зі змінною Y, і якщо так, то в чому цей зв'язок полягає, і чи можемо ми його використати для того, щоб передбачити Y?

Проста лінійна регресія, або парна лінійна регресія, моделює зв'язок між величиною однієї змінної та величиною другої, наприклад у міру збільшення X збільшується і Y. Або ж у міру збільшення X зменшується і Y. Кореляція – ще один спосіб виміряти те, яким чином дві змінні зв'язані між собою. Різниця між ними полягає в тому, що кореляція вимірює силу зв'язку між двома змінними, тоді як регресія оцінює природу зв'язку кількісно.

2.1.2 Рівняння регресії

Проста лінійна регресія оцінює, наскільки саме зміниться Y коли X змінюється на деяку величину. Для коефіцієнта кореляції змінні X та Y взаємозамінні. У разі регресії ми намагаємося передбачити змінну Y за змінною X, використовуючи лінійний зв'язок (тобто прямий):

$$Y = b_0 + b_1 X. \qquad (2.1)$$

Ця формула читається, як «Y дорівнює b_1, помножене на X плюс константа b_0». Компонент рівняння b_0 називається перетином (або константою), а b_1 – нахилом по відношенню до осі x. Змінна Y називається *відгуком* або *залежною змінною*, оскільки вона залежить від X. Змінна X називається *провісником* (предиктором, від англ. predictor), або *незалежною змінною*. Спільнота машинного навчання тяжіє до використання інших термінів, називаючи Y *ціллю* та X – *вектором ознак*.

2.1.3 Найменші квадрати

Яким чином виконується підгонка моделі до даних? Коли існує чіткий зв'язок, ви можете подумки уявити підгонку прямої вручну. Насправді пряма

регресії є оцінкою, яка мінімізує значення суми квадратичних залишків, також іменованих *сумою квадратів залишків* чи *залишковою сумою квадратів* (residual sum of squares, RSS):

$$RSS = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 = \sum_{i=1}^{n} \left(Y_i - \hat{b}_0 - \hat{b}_1 X_i\right)^2 \qquad (2.2)$$

Оцінки $\hat{b}_0$ та $\hat{b}_1$ – це значення, які мінімізують суму квадратів залишків (RSS).

Метод мінімізації суми квадратів залишків називається *регресією на основі найменших квадратів*, чи регресією *на основі звичайних найменших квадратів* (ЗНК). Цей метод часто приписується Карлу Фрідріху Гауссу, німецькому математику, але він був вперше опублікований в 1805 французьким математиком Андре-Марі Лежандром. Регресію на основі найменших квадратів можна легко та швидко обчислити за допомогою стандартної статистичної обчислювальної системи.

Історично, обчислювальна зручність найменших квадратів є однією з причин широкого застосування цього методу в регресії. З появою великих даних обчислювальна швидкість, як і раніше, залишається важливим фактором. Найменші квадрати, як і середнє значення, як метод чутливий до викидів, хоча цей факт має тенденцію бути значною проблемою тільки в малих або помірних за розміром наборах даних.

З появою великих даних регресія широко використовується для формування моделі з метою передбачення індивідуальних наслідків для нових даних замість статистичного пояснення даних, наявних під рукою (тобто передбачувальної моделі). У цьому випадку головними елементами, що цікавлять, є підігнані значення $\hat{Y}$.

У маркетингу регресія може використовуватися для передбачення зміни в доході у відповідь на розмір рекламної кампанії. Університети використовують регресію для передбачення середнього академічного бала GPA студентів на основі їхніх балів за іспит на визначення академічних здібностей SAT.

Регресійна модель, яка підігнана до даних добре, налаштована так, що зміни в X призводять до змін в Y. Однак саме по собі рівняння регресії не доводить напрямок причинно-наслідкової обумовленості. Висновки про причинно-наслідкову обумовленість слід робити, виходячи з ширшого контексту розуміння зв'язку. Наприклад, рівняння регресії могло б показати певний зв'язок між числом натискань на веб-рекламі та числом конверсій. Саме наше знання маркетингового процесу, а не рівняння регресії приводить нас до висновку про те, що натискання на рекламі генерують продажі, і не навпаки.

2.1.4 Множинна лінійна регресія

Коли провісників кілька, рівняння регресії просто розширюється їх розміщення:

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_pX_p + e. \quad (2.3)$$

Замість прямої тепер у нас лінійна модель – зв'язок між кожним коефіцієнтом та його змінною (ознакою) є лінійним.

Всі інші поняття з простої лінійної регресії, такі як підгонка найменшими квадратами, підігнані значення та залишки, відносяться і до умов множинної лінійної регресії. Наприклад, підігнані значення задаються наступною формулою:

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1X_{1,i} + \hat{b}_2X_{2,i} + \cdots + \hat{b}_pX_{p,i}. \quad (2.4)$$

2.1.5 Перехресний контроль

Усі класичні статистичні регресійні метрики є «внутрішньовибірковими» метриками – вони застосовуються до тих самих даних, які використовувалися для підгонки моделі. Інтуїтивно ви розумієте, що буде цілком логічним відкладати трохи вихідних даних, не використовуючи їх для підгонки моделі, а потім застосовувати модель до відкладених даних, щоб побачити, як добре вона справляється зі своєю роботою. Зазвичай ви будете використовувати більшу частину даних для підгонки моделі, а решту – для її тестування.

Ідея перевірки поза вибіркою не є новою, але вона не утвердилася до тих пір, поки великі набори даних не стали більш переважаючими; маючи в розпорядженні малий набір даних, аналітики, як правило, хочуть використовувати всі наявні дані і на їх основі виконувати підгонку кращої моделі.

Використання відкладеної вибірки ставить вас у залежність від деякої невизначеності, що виникає просто через варіабельність у малій відкладеній вибірці. Наскільки відрізнятимуться результати аналізу моделі, якби ви отримували іншу відкладену вибірку?

Перехресний контроль розширює ідею відкладеної вибірки до послідовних відкладених вибірок.

Псевдокод базового k-блочного перехресного контролю

1. Відкласти $1/k$ даних як відкладену вибірку
2. Натренувати модель на даних, що залишилися.
3. Застосувати модель до відкладеної вибірки $1/k$ (виставити їй бал) та записати необхідні метрики оцінювання результативності моделі.
4. Відновити перші $1/k$ даних і відкласти наступне $1/k$ (за винятком будь-яких записів, які були обрані вперше).
5. Повторити кроки 2-4.

6. Повторювати доти, доки кожен запис не буде використаний у відкладеній частці.
7. Усереднити чи іншим чином скомбінувати метрики аналізу моделі.

Розподіл даних на тренувальну та відкладену вибірки також називається розподілом на блоки.

2.2 Приклад розв’язування задачі

2.2.1 Приклад: визначення впливу бавовняного пилу на легені

Розглянемо діаграму розсіювання на рис. 2.1, що показує число років, протягом яких робітник зазнав впливу бавовняного пилу (Exposure) проти показника об'єму легень (PEFR – «пікова об'ємна швидкість видиху»). Яким чином змінна PEFR пов'язана з Exposure? Важко сказати щось конкретне, просто дивлячись на зображення.

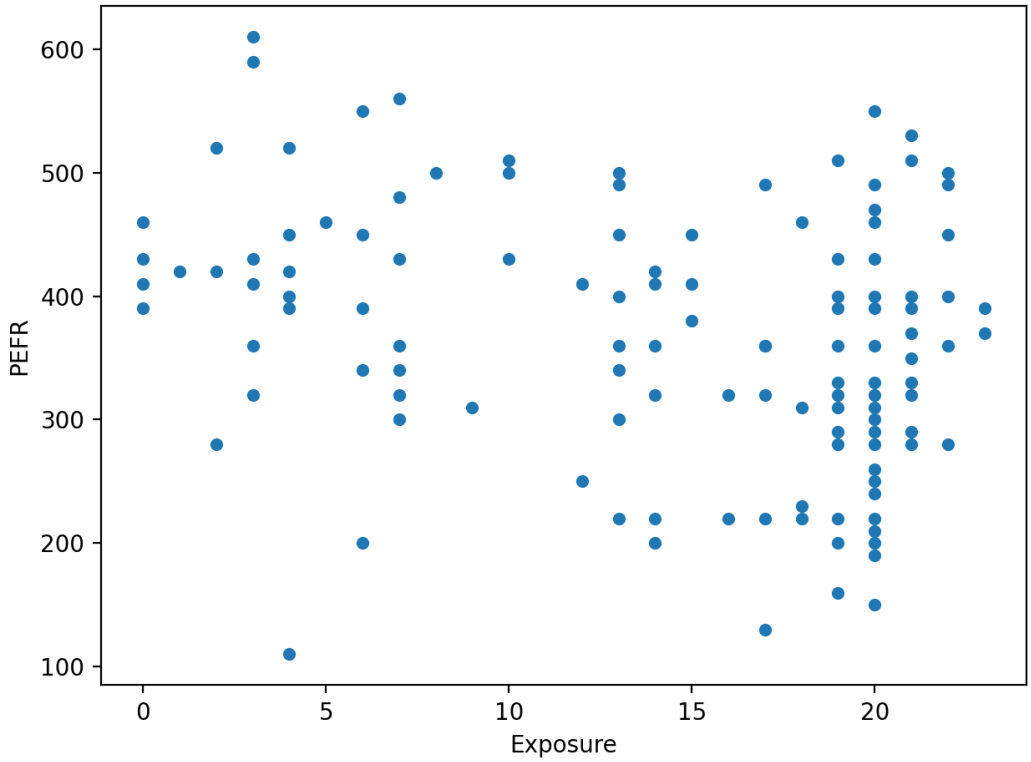

Рис. 2.1 – Діаграма розсіювання пікової об'ємної швидкість видиху.

Код

```
import pandas as pd
import matplotlib.pyplot as plt

lung = pd.read_csv('LungDisease.csv')

lung.plot.scatter(x='Exposure', y='PEFR')
```

```
plt.show()
```

Проста лінійна регресія намагається відшукати «оптимальну» пряму, щоб передбачити відгук PEFR, як функцію від передбачувальної змінної Exposure:

$$\text{PEFR} = b_0 + b_1 Exposure, \tag{2.5}$$

де b_0 – перетин, b_1 – коефіцієнт регресії.

Для навчання лінійної регресії в Python використовується клас LinearRegression з бібліотеки scikit-learn. Для навчання моделі використовується метод fit, в який передаються незалежні змінні (вектор ознак) та залежна змінна навчальної вибірки. Для передбачення використовується метод predict.

Код

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression


lung = pd.read_csv('LungDisease.csv')

lung.plot.scatter(x='Exposure', y='PEFR')

plt.tight_layout()
plt.show()

predictors = ['Exposure']
outcome = 'PEFR'

model = LinearRegression()
model.fit(lung[predictors], lung[outcome])

print('Intercept')
print(model.intercept_)
print('Coefficient Exposure')
print(model.coef_[0])

lung.plot.scatter(x='Exposure', y='PEFR')
plt.plot(lung['Exposure'], model.predict(lung[predictors]))

plt.show()
```

Результат навчання моделі показано на рис. 2.2. Тепер стає очевидно, що пікова об’ємна швидкість видиху залежить від кількості років вдихання бавовняного пилу на фабриці.

Отримані коефіцієнти становлять: константа: 424.583, нахил: -4.185.

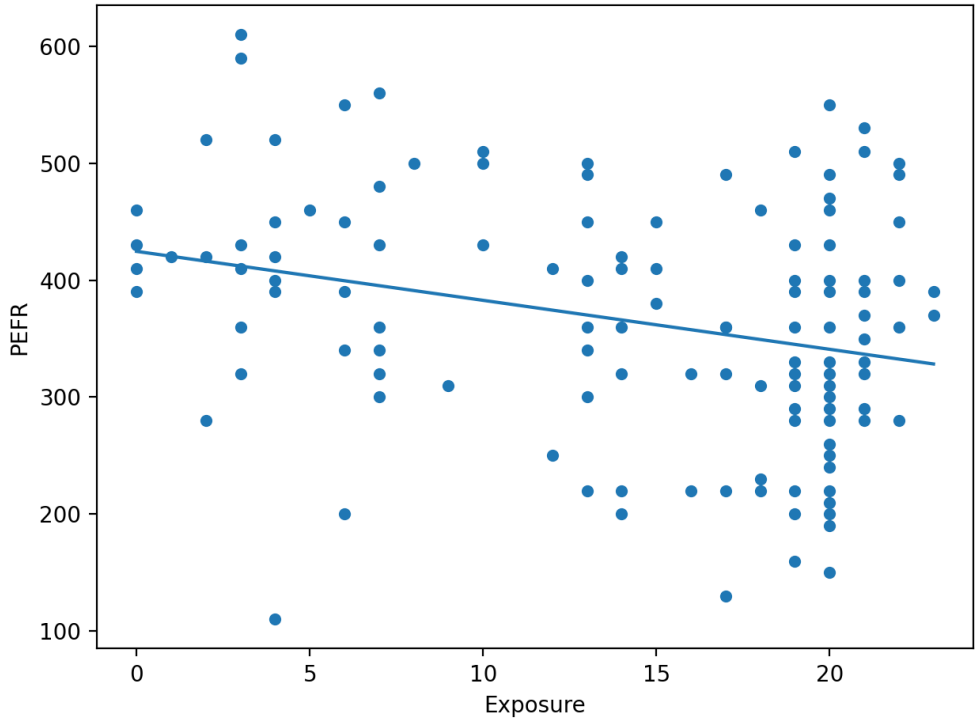

Рис. 2.2 – Підігнана пряма регресії.

2.2.2 Створення нелінійних ознак

Розглянемо наступний набір даних:

data.csv

```
#,x,y
0,-5.0,26.697156
1,-4.0,16.989532
2,-3.0,9.098270
3,-2.0,4.513944
4,-1.0,1.207257
5,0.0,0.193051
6,1.0,1.045055
7,2.0,4.582720
8,3.0,9.152186
9,4.0,16.385336
```

Зробимо візуалізацію отриманих даних. Поглянувши на рис. 2.3 стає очевидно, що залежність y від x має нелінійний характер. В такому випадку лінійна регресія дасть високу помилку передбачення та не буде коректно відображати зв’язок між y та x.

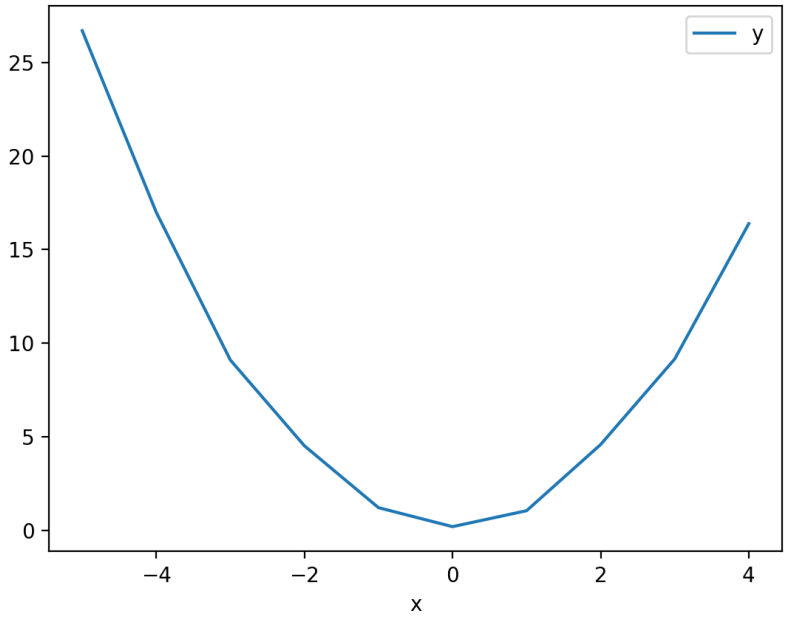

Рис. 2.3 – Візуалізація набору даних.

Помітимо, що в бібліотеці scikit-learn немає окремих класів для навчання квадратичної, кубічної або будь-якої нелінійної регресії. На то є причина – якщо навчати лінійну регресію на попередньо створених нелінійних ознаках – отримаємо нелінійну регресію!

Створимо в pandas додатковий стовпчик із нелінійною ознакою наступним чином та навчимо лінійну регресію:

Код

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

df = pd.read_csv('data.csv')

predictors = ['x', 'x_squared']
outcome = 'y'

df['x_squared'] = df['x'] * df['x']
lr = LinearRegression()
lr.fit(df[predictors], df[outcome])

print('Intercept')
print(lr.intercept_)
print('Coefficients')
for name, value in zip(predictors, lr.coef_):
    print(name, value)
```

Вивід

```
Intercept
```

```
0.10809277575757115
Coefficients
x -0.06002693560606039
x_squared 1.0409817007575763
```

Оскільки справжня залежність має вигляд y = x² + шум, коефіцієнт для x близький до 0 (він нам не потрібен), а коефіцієнт для x_squared близький до 1. Перетин з віссю Y близький до 0 (оскільки y=0, коли x=0). Прогнози (y_pred) майже ідеально збігаються з фактичними значеннями y (відмінності зумовлені шумом). **Важливо.** Без x_squared модель не працювала б (наприклад, y = x була б прямою лінією, а не кривою).

Лінійна регресія може моделювати тільки прямолінійні залежності. Додавання таких ознак, як x^2, x^3 або $x \cdot y$, дозволяє їй фіксувати криві (квадратичні, кубічні) або взаємодії.

2.3 Індивідуальне завдання

Лабораторна робота присвячена регресійному аналізу даних. Робота виконується із використанням мови програмування Python та бібліотек Pandas, sklearn, NumPy, Matplotlib. За бажанням студента, допускається виконання роботи мовою програмування R.

1. Зібрати дані відповідно до власного варіанту (табл. на наступній сторінці). Має бути якнайменше 20 записів в даних. Навести дані у звіті із розширеною інформацією (в залежності від варіанта: країни, міста, марки автомобілів). Зазначити з яких сайтів було зібрано дані.
2. Розбити вибірку на тренувальну та валідаційну (пропорція розбиття відповідно до варіанту).
3. Навчити лінійну регресію на тренувальній та оцінити на валідаційній вибірці.
4. Провести крос-валідацію за кількістю блоків відповідно до варіанта. В функцію для крос-валідації необхідно подавати всю вибірку.
5. Створити квадратичні та кубічні ознаки.
6. Розрахувати оцінку на валідаційній вибірці та із використанням крос-валідації для кожного типу регресії. Результати валідації та крос-валідації різних типів регресій необхідно зібрати в єдину таблицю. Обрати найкращу модель за крос-валідаційною вибіркою.
7. Виписати функцію для кожного типу регресії (із коефіцієнтами та константою).
8. Зобразити початкові дані та всі 3 кривих регресії на одному графіку. Подумайте які типи графіків слід використати.
9. Застосувати кращу модель до тестової вибірки.

Робота має бути представлена у вигляді звіту (файл pdf). Звіт має містити титульний аркуш, номер варіанта, розрахунки, необхідні описи (відповідно до завдання вище) та вихідний код програм.

Назва файлу: Прізвище_група

Роботи, виконані не за варіантом, не приймаються.

№	Галузь	Дані	Залежність	Параметри перевірки 1. Розбиття (тренування валідація) 2. Кількість блоків для крос-валідації 3. Оцінка для вибору кращої моделі	Тестові дані
1	Пасажирські авіаперевезення	Дальність і час перельоту між різними містами	Час від дальності	1. Розбиття 80/20 2. 3 блоки 3. Середньоквадратична похибка	Час польоту на 500, 1000 і 3000 км
2	Пасажирські авіаперевезення	Дальність і вартість перельоту між різними містами економ-класом	Вартість від дальності	1. Розбиття 75/25 2. 3 блоки 3. Середня абсолютна похибка	Вартість перельоту на 500, 1000 і 3000 км
3	Пасажирські авіаперевезення	Дальність і вартість перельоту між різними містами бізнес-класом	Вартість від дальності	1. Розбиття 85/15 2. 3 блоки 3. Коефіцієнт детермінації	Вартість перельоту на 500, 1000 і 3000 км
4	Пасажирські залізничні перевезення	Дальність і час поїздки між різними містами	Час від дальності	1. Розбиття 70/30 2. 3 блоки 3. Середньоквадратична похибка	Час поїздки на 400, 8000 і 2000 км
5	Пасажирські залізничні перевезення	Дальність і вартість поїздки між різними містами в плацкарті	Вартість від дальності	1. Розбиття 80/20 2. 5 блоків 3. Середня абсолютна похибка	Вартість поїздки на 400, 8000 і 2000 км
6	Пасажирські залізничні перевезення	Дальність і вартість поїздки між різними містами в купе	Вартість від дальності	1. Розбиття 75/25 2. 5 блоків 3. Коефіцієнт детермінації	Вартість поїздки на 400, 8000 і 2000 км
7	Ринок нерухомості	Площа і вартість квартир на первинному ринку	Вартість від площі	1. Розбиття 85/15 2. 5 блоків	Вартість для площі 30, 50, 100 кв.м

				3. Середньоквадратична похибка	
8	Ринок нерухомості	Площа і вартість квартир на вторинному ринку	Вартість від площі	1. Розбиття 70/30 2. 5 блоків 3. Середня абсолютна похибка	Вартість для площі 30, 50, 100 кв.м
9	Ринок автотранспорту	Вартість і пробіг автомобілів певної марки на вторинному ринку	Вартість від пробігу	1. Розбиття 80/20 2. 3 блоки 3. Коефіцієнт детермінації	Вартість для пробігу 20 тис., 50 тис, 150 тис. км
10	Ринок автотранспорту	Вартість і вік автомобілів певної марки на вторинному ринку	Вартість від віку	1. Розбиття 75/25 2. 3 блоки 3. Середньоквадратична похибка	Вартість від віку 2 роки, 5 років, 10 років
11	Ринок автотранспорту	Вік і пробіг автомобілів певної марки на вторинному ринку	Пробіг від віку	1. Розбиття 85/15 2. 3 блоки 3. Середня абсолютна похибка	Пробіг для віку 2 роки, 5 років, 10 років
12	Світова економіка	Тривалість життя та доходи на душу населення країн світу	Тривалість життя від доходів	1. Розбиття 70/30 2. 3 блоки 3. Коефіцієнт детермінації	Тривалість життя для доходів 5,20, 50 тис.дол.
13	Пасажирські авіаперевезення	Дальність і час перельоту між різними містами	Час від дальності	1. Розбиття 80/20 2. 5 блоків 3. Середня абсолютна похибка	Час польоту на 500, 1000 і 3000 км
14	Пасажирські авіаперевезення	Дальність і вартість перельоту між різними містами економ-класом	Вартість від дальності	1. Розбиття 75/25 2. 5 блоків 3. Коефіцієнт детермінації	Вартість перельоту на 500, 1000 і 3000 км
15	Пасажирські авіаперевезення	Дальність і вартість перельоту між різними містами бізнес-класом	Вартість від дальності	1. Розбиття 85/15 2. 5 блоків 3. Середньоквадратична похибка	Вартість перельоту на 500, 1000 і 3000 км
16	Пасажирські залізничні перевезення	Дальність і час поїздки між різними містами	Час від дальності	1. Розбиття 70/30 2. 5 блоків 3. Середня абсолютна похибка	Час поїздки на 400, 8000 і 2000 км

17	Пасажирські залізничні перевезення	Дальність і вартість поїздки між різними містами в плацкарті	Вартість від дальності	1. Розбиття 80/20 2. 3 блоки 3. Коефіцієнт детермінації	Вартість поїздки на 400, 8000 і 2000 км
18	Пасажирські залізничні перевезення	Дальність і вартість поїздки між різними містами в купе	Вартість від дальності	1. Розбиття 75/25 2. 3 блоки 3. Середньоквадратична похибка	Вартість поїздки на 400, 8000 і 2000 км
19	Ринок нерухомості	Площа і вартість квартир на первинному ринку	Вартість від площі	1. Розбиття 85/15 2. 3 блоки 3. Середня абсолютна похибка	Вартість для площі 30, 50, 100 кв.м
20	Ринок нерухомості	Площа і вартість квартир на вторинному ринку	Вартість від площі	1. Розбиття 70/30 2. 3 блоки 3. Коефіцієнт детермінації	Вартість для площі 30, 50, 100 кв.м
21	Ринок автотранспорту	Вартість і пробіг автомобілів певної марки на вторинному ринку	Вартість від пробігу	1. Розбиття 80/20 2. 5 блоків 3. Середньоквадратична похибка	Вартість для пробігу 20 тис., 50 тис, 150 тис. км
22	Ринок автотранспорту	Вартість і вік автомобілів певної марки на вторинному ринку	Вартість від віку	1. Розбиття 75/25 2. 5 блоків 3. Середня абсолютна похибка	Вартість від віку 2 роки, 5 років, 10 років
23	Ринок автотранспорту	Вік і пробіг автомобілів певної марки на вторинному ринку	Пробіг від віку	1. Розбиття 85/15 2. 5 блоків 3. Коефіцієнт детермінації	Пробіг для віку 2 роки, 5 років, 10 років
24	Світова економіка	Тривалість життя та доходи на душу населення країн світу	Тривалість життя від доходів	1. Розбиття 70/30 2. 5 блоків 3. Середньоквадратична похибка	Тривалість життя для доходів 5,20, 50 тис.дол.
25	Ринок нерухомості	Площа і вартість квартир на первинному ринку	Вартість від площі	1. Розбиття 85/15 2. 3 блоки 3. Коефіцієнт детермінації	Вартість для площі 30, 50, 100 кв.м
26	Ринок нерухомості	Площа і вартість квартир на вторинному ринку	Вартість від площі	1. Розбиття 70/30 2. 3 блоки 3. Середньоквадратична похибка	Вартість для площі 30, 50, 100 кв.м

27	Ринок автотранспорту	Вартість і пробіг автомобілів певної марки на вторинному ринку	Вартість від пробігу	1. Розбиття 80/20 2. 5 блоків 3. Середня абсолютна похибка	Вартість для пробігу 20 тис., 50 тис, 150 тис. км
28	Ринок автотранспорту	Вартість і вік автомобілів певної марки на вторинному ринку	Вартість від віку	1. Розбиття 75/25 2. 5 блоків 3. Коефіцієнт детермінації	Вартість від віку 2 роки, 5 років, 10 років
29	Ринок автотранспорту	Вік і пробіг автомобілів певної марки на вторинному ринку	Пробіг від віку	1. Розбиття 85/15 2. 5 блоків 3. Середньоквадратична похибка	Пробіг для віку 2 роки, 5 років, 10 років
30	Пасажирські залізничні перевезення	Дальність і вартість поїздки між різними містами в купе	Вартість від дальності	1. Розбиття 75/25 2. 5 блоків 3. Середня абсолютна похибка	Вартість поїздки на 400, 8000 і 2000 км

2.4 Контрольні питання

1. Яке рівняння лінійної регресії?
2. Що означають $Y, \hat{Y}, \hat{b}_0, \hat{b}_1$ в рівнянні регресії? Для чого використовується позначка «^»?
3. Який функціонал мінімізується задля знаходження коефіцієнтів $\hat{b}_0, \hat{b}_1$?
4. Що таке відгук, незалежна змінна, перетин, залишки в регресії?
5. Чим відрізняється множинна лінійна регресія від простої лінійної регресії?
6. Які метрики ви знаєте для оцінки якості регресії?
7. Що таке перехресний контроль? Як він дозволяє оцінити якість певного алгоритму на нових даних?
8. Для k-блочного перехресного контролю на одній ітерації скільки блоків використовується для навчання, а скільки для перевірки якості?

Практична робота №3 – Класифікація за допомогою логістичної регресії та наївного Баєса

Мета роботи: закріплення навичок роботи із категорійними даними, їх попередньої обробки та класифікація даних.

Практична робота присвячена роботі з факторними змінними, вирішенню задачі класифікації декількома методами, оцінці точності вирішення задачі. Робота виконується із використанням мови програмування Python та бібліотек Pandas, SciPy, NumPy, Matplotlib. За бажанням студента, допускається виконання роботи мовою програмування R. Для розв'язання задачі практичної роботи за узгодженням з викладачем студент може запропонувати свій набір даних.

3.1 Теоретичні відомості

3.1.1 Як працювати із категорійними даними (факторними змінними)?

Факторні змінні, іменовані також категоріальними змінними, приймають граничне число дискретних значень. Наприклад, метою позики може бути «консолідація заборгованості», «весілля», «автомобіль» тощо. Двійкова (так/ні) змінна, іменована також індикаторною змінною, є особливим випадком факторної змінної. Регресія вимагає на вході числові дані, тому факторні змінні потрібно перекодувати, щоб їх можна було використовувати в моделі. Підхід, який найчастіше зустрічається, полягає в конвертуванні змінної в множину двійкових фіктивних змінних

Фіктивні змінні – це двійкові змінні, що приймають значення 0 і 1 і виводяться шляхом перекодування факторних даних для використання в регресії та інших моделях.

Опорне кодування – тип кодування в якому один рівень фактора вибирається як опорний, а інші фактори зіставляються із цим рівнем.

Кодувальник з одним активним станом – тип кодування, загальноприйнятий у співтоваристві машинного навчання, у якому зберігаються всі рівні чинників. Широко використовується в деяких алгоритмах самонавчання; водночас цей прийом не підходить для множинної лінійної регресії.

Девіаційне кодування – тип кодування, при якому кожен рівень порівнюється не з опорним рівнем, а з сукупним середнім.

3.1.2 Приклад: дані житлового фонду округу Кінг

У даних житлового фонду округу Кінг є факторна змінна, що відповідає типу власності; нижче показано малу підмножину із шести записів. Є три можливих значення: Multiplex, Single Family і Townhouse. Для того щоб скористатися зазначеною факторною змінною, ми маємо конвертувати її в множину двійкових змінних. Це робиться шляхом створення двійкової змінної для кожного можливого значення факторної змінної.

Табл. 3.1. Дані про тип власності.

№	PropertyType
1	Multiplex
2	Single Family
3	Single Family
4	Single Family
5	Single Family
6	Townhouse

Кодувальник з одним активним станом. У Python ми можемо конвертувати категоріальні змінні у фіктивні за допомогою методу get_dummies пакета pandas. За замовчуванням метод повертає кодування категоріальної змінної з одним активним станом:

Код

```
import pandas as pd

pd.get_dummies(house['PropertyType']).head()
```

Табл. 3.2. Перетворені дані про тип власності за допомогою кодувальника з єдиним активним станом.

№	PropertyTypeMultiplex	PropertyTypeSingleFamily	PropertyTypeTownhouse
1	1	0	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	1	0
6	0	0	1

Опорне кодування. Іменований аргумент drop_first повертатиме $P - 1$ стовпців. Використовуйте його, щоб уникнути проблеми мультиколінеарності:

Код

```
import pandas as pd

pd.get_dummies(house['PropertyType'], drop_first=True).head()
```

У деяких алгоритмах, що автоматично навчаються, як-от найближчі сусіди і моделі на основі дерев рішень, кодування з одним активним станом є стандартним способом представлення факторних змінних. У регресійному формулюванні факторна змінна з P чітко помітними рівнями зазвичай подається

матрицею тільки з $P-1$ стовпчиками. Це зумовлено тим, що регресійна модель у типовій ситуації включає член перетину. Говорячи про перетин, після того як ви визначили значення для $P-1$ двійкових стовпчиків, значення P-го стає відомим і може вважатися надлишковим. Додавання P-го стовпця викличе помилку мультиколінеарності.

Табл. 3.3. Перетворені дані про тип власності за допомогою опорного кодування.

№	PropertyTypeSingleFamily	PropertyTypeTownhouse
1	0	0
2	1	0
3	1	0
4	1	0
5	1	0
6	0	1

3.1.3 Упорядковані факторні змінні

Деякі факторні змінні відображають рівні фактора. Вони називаються впорядкованими факторними змінними або впорядкованими категоріальними змінними. Наприклад, категорія якості позики може бути A, B, C тощо. – кожна категорія несе в собі більший ризик, ніж попередня категорія. Нерідко впорядковані факторні змінні можуть бути конвертовані в числові значення і використовуватися як є. Наприклад, змінна «клас будинка» – це впорядкована факторна змінна. Кілька типів категорій якості наведено в табл. 3.4. Хоча категорії якості мають конкретний сенс, числове значення впорядковано від низу до верху, відповідаючи будинкам вищої якості. Розгляд упорядкованих чинників як числової змінної зберігає інформацію, що міститься в упорядкуванні, яка буде втрачена, якщо його конвертувати у фактор.

Табл. 3.4. Приклад упорядкованих факторних змінних.

Значення	Опис
1	Низькобюджетне
2	Нижче середнього
5	Задовільне
10	Дуже гарне
12	Розкішне
13	Особняк

3.1.4 Поняття класифікації даних

Класифікація – це керований метод машинного навчання, коли модель намагається передбачити правильну мітку для заданих вхідних даних. При класифікації модель повністю навчається на навчальних даних, а потім оцінюється на тестових даних перед тим, як використовувати її для прогнозування на нових, ще не бачених даних.

Наприклад, алгоритм може навчитися передбачати, чи є даний електронний лист спамом або ні, як показано на рис. 3.1.

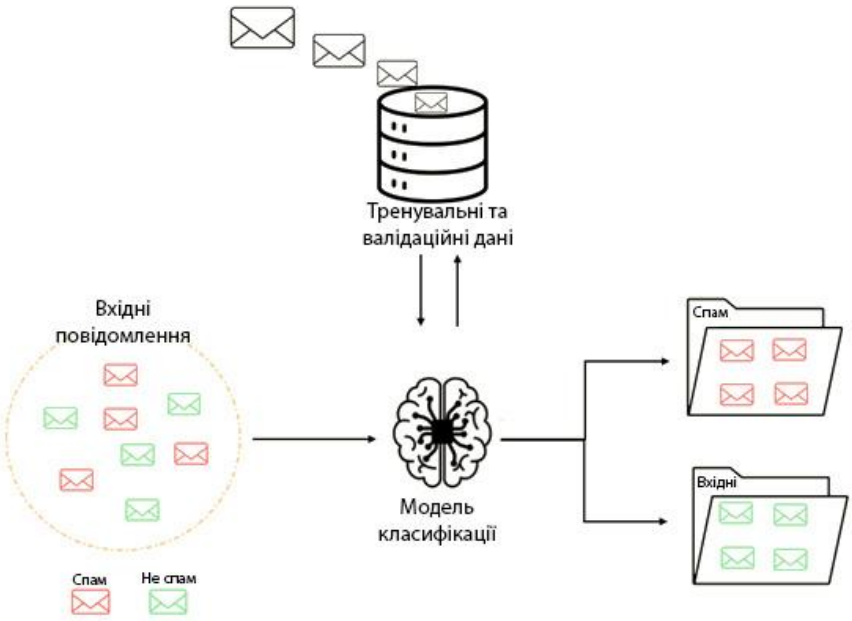

Рис. 3.1. Вирішення спам/не спам – один з прикладів задачі класифікації.

Перш ніж зануритися в концепцію класифікації, ми спочатку зрозуміємо різницю між двома типами учнів у класифікації: лінивими та охочими до навчання. Потім ми прояснимо помилкове уявлення про класифікацію та регресію.

3.1.5 Ліниві учні та учні, які прагнуть вчитися

У класифікації машинного навчання існує два типи учнів: ліниві та активні учні.

Активні учні – це алгоритми машинного навчання, які спочатку будують модель на основі навчального набору даних, перш ніж робити будь-які прогнози щодо майбутніх наборів даних. Вони витрачають більше часу на процес навчання через своє бажання отримати краще узагальнення під час навчання від вивчення вагових коефіцієнтів, але їм потрібно менше часу, щоб робити прогнози.

Більшість алгоритмів машинного навчання – це активні учні – алгоритми, що швидко навчаються, і нижче наведено кілька прикладів:

- логістична регресія;
- метод опорних векторів;
- дерева рішень;
- штучні нейронні мережі.

З іншого боку, ліниві учні або учні на основі прикладів не створюють жодної моделі одразу на основі навчальних даних, і саме звідси походить їхня лінивість. Вони просто запам'ятовують навчальні дані, і кожного разу, коли виникає необхідність зробити прогноз, вони шукають найближчого сусіда з усіх навчальних даних, що робить їх дуже повільними під час прогнозування. Деякі приклади такого роду:

- метод K-найближчих сусідів;
- міркування на основі конкретних випадків.

3.1.6 Класифікація в машинному навчанні проти регресії

Хоча класифікація і регресія відносяться до категорії керованого навчання, вони не однакові.

- Завданням прогнозування є *класифікація*, коли цільова змінна є дискретною. Прикладом може слугувати ідентифікація основної думки, що лежить в основі тексту.
- Завданням прогнозування є *регресія*, коли цільова змінна є неперервною. Прикладом може бути прогнозування заробітної плати людини, враховуючи її освіту, попередній досвід роботи, географічне розташування та рівень стажу.

3.1.7 Логістична регресія

Незважаючи на свою назву, логістична регресія є, по суті, алгоритмом класифікації, а не методом регресії. Її сила полягає в простоті, інтерпретованості та ефективності для бінарної класифікації (двох класів), хоча вона природно поширюється і на багатокласові задачі. Ключовим моментом є те, що лінійна регресія, яка видає безперервне значення, не підходить для прогнозування дискретних категорій. Логістична регресія вирішує цю проблему шляхом застосування логістичної функції (також званої сигмоїдної функції) до результату лінійної моделі.

Уявіть простий набір даних, де ми хочемо передбачити, чи складе студент іспит (1) або провалить його (0) на основі кількості годин, присвячених навчанню. Лінійна модель може передбачити значення 2,5 для 5 годин навчання. Але це значення не відповідає ймовірності складання іспиту. Логістична функція $\sigma(z) = \frac{1}{1+e^{-z}}$ бере будь-яке дійсне число z (наприклад, вихідні дані лінійної моделі $z = w_0 + w_1 \cdot x_1 + ... + w_n \cdot x_n$) і відображає його на значення від 0 до 1. Цей результат інтерпретується як ймовірність того, що точка даних належить до позитивного класу (наприклад, «склав»). Ймовірність вище 0,5 зазвичай класифікується як 1 («склав»), нижче 0,5 – як 0 («не склав»).

Модель вивчає коефіцієнти $(w_0, w_1, \ldots, w_n)$, які максимізують ймовірність спостереження фактичних міток класів у навчальних даних, з огляду на прогнозовані ймовірності. Це часто досягається за допомогою техніки оптимізації, такої як градієнтний спуск.

3.1.8 Метод опорних векторів

Метод опорних векторів (SVM) використовує інший, геометрично інтуїтивний підхід до класифікації. Замість прямого моделювання ймовірностей, SVM шукає оптимальну гіперплощину, яка найкраще розділяє класи в просторі ознак. Оптимальна гіперплощина визначається як та, що максимізує відстань між гіперплощиною та найближчими точками даних кожного класу, відомими як опорні вектори.

Уявіть простий 2D-набір даних з двома класами. SVM знаходить лінію (гіперплощину), яка розділяє дві групи точок, забезпечуючи максимально можливий проміжок між лінією та найближчими точками кожного класу. Точки, що знаходяться не з того боку лінії або занадто близько до лінії (в межах відстані), є потенційними опорними векторами. Мета SVM – максимізувати цю відстань, оскільки більша відстань, як правило, означає краще узагальнення для невидимих даних.

Ключовою є функція завісних втрат, яка карає помилкові класифікації та точки в межах межі. Для лінійно роздільних даних SVM знаходить ідеальну роздільну гіперплощину. Для нелінійно роздільних даних (що майже завжди має місце) SVM використовують ядерний трюк, щоб неявним чином відобразити оригінальні ознаки у вищий вимірний простір, де можна знайти лінійний роздільник. Поширені ядра включають лінійні (без відображення), поліноміальні та rbf (радіальна базова функція, найпопулярніша для нелінійних задач, яка створює плавні, гнучкі межі).

3.1.9 Наївний Баєс

Наївні класифікатори Баєса базуються на теоремі Баєса, фундаментальній концепції теорії ймовірностей. Теорема Баєса дозволяє обчислити апостеріорну ймовірність класу за певних ознак, виходячи з апріорної ймовірності класу та ймовірності ознак за умови класу. «Наївність» полягає в сильному, часто нереалістичному припущенні: що всі ознаки є умовно незалежними за умови ярлика класу.

Теорема Баєса стверджує:

$$P(\text{Клас} \mid \text{Ознаки}) = \frac{P(\text{Ознаки}|\text{Клас}) \cdot P(\text{Клас})}{P(\text{Ознаки})} \qquad (3.1)$$

Класифікатор оцінює $P(\text{Клас} \mid \text{Ознаки})$ для кожного класу. Знаменник $P(\text{Ознаки})$ є постійним для всіх класів і може бути проігнорований для класифікації (нам просто потрібно знайти клас, що максимізує чисельник). Ключовим спрощенням є наївне припущення про незалежність:

$$P(\text{Ознаки}|\text{Клас}) = P(\text{Ознака}_1|\text{Клас}) \cdot P(\text{Ознака}_2|\text{Клас}) \cdot \ldots \cdot P(\text{Ознака}_N|\text{Клас}) \quad (3.2)$$

Це припускає, що знання значення однієї ознаки нічого не говорить нам про значення іншої ознаки, враховуючи клас. Це рідко буває правдою в реальних даних (наприклад, розмір і вага фрукта корелюються), але, що примітно, наївний Баєс часто працює напрочуд добре, незважаючи на це припущення.

3.2 Приклад розв'язування задачі

У світі бізнес-аналітики рішення про затвердження проектів залежать від багатьох взаємопов'язаних факторів. В даній задачі необхідно навчити модель класифікації, що визначатиме статус для нового проєкту, використовуючи наявні його характеристики. Самі характеристики представлено:

- чисельними ознаками;
- категорійними впорядкованими;
- категорійними невпорядкованими змінними.

Крок 1. Імпорт необхідних бібліотек та огляд даних:

Код

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import cross_val_score

df = pd.read_csv('30.csv')
print(df.head())
```

Набір даних містить 8 стовпців, включаючи цільову змінну «Статус проєкту», яка містить категоріальні значення, такі як «Узгоджено» та «Відхилено».

Крок 2. Визначення категорійних змінних та їх кодування.

Категорійні дані не можуть бути подані в модель як є та мають бути закодовані:

- невпорядковані – використовуючи опорне кодування. Для цього використати: `pd.get_dummies(df, drop_first=True)`;
- впорядковані – числами 0, 1, 2… (реалізувати самостійно)

Крок 3. Навчання логістичної регресії та її оцінка на крос валідації.

Зауважу, що в прикладі навчання здійснено із використанням лише однієї ознаки – «Лояльність клієнта». В практичній роботі необхідно використати всі наявні ознаки (закодовані відповідно до типу ознаки на кроці 2) для передбачення єдиної цільової змінної – «Статус проєкта».

Код

```
predictors = ['Лояльність клієнта']
outcome = 'Статус проєкта_Узгоджено'
m = LogisticRegression()
m.fit(df[predictors], df[outcome])

print(cross_val_score(m, df[predictors], df[outcome], cv=5, scoring='accuracy'))

m = GaussianNB()
m.fit(df[predictors], df[outcome])

print(cross_val_score(m, df[predictors], df[outcome], cv=5, scoring='accuracy'))
```

Крок 4. Навчання моделі наївного Баєса та її оцінка на крос валідації.

Код

```
m = GaussianNB()
m.fit(df[predictors], df[outcome])

print(cross_val_score(m, df[predictors], df[outcome], cv=5, scoring='accuracy'))
```

3.3 Індивідуальне завдання

1. Використати дані відповідно до власного варіанта (посилання на дані доступне на порталі дистанційної освіти). Вивести таблицю за допомогою Pandas.
2. Визначити числові, категорійні впорядковані, категорійні невпорядковані змінні. Для кожної категорійної змінної написати код, який виводить всі унікальні значення, котрі вона набуває в наборі даних.
3. Закодувати категорійні впорядковані числами 0, 1, 2… (або іншими числами аналогічним чином), до невпорядкованих застосувати опорне кодування.
4. Застосувати методи логістичної регресії та наївного Баєса до оброблених даних для передбачення змінної «Статус проєкта». На крос-валідації обрати кращу модель.

Робота має бути представлена у вигляді звіту (файл pdf). Звіт має містити титульний аркуш, номер варіанта, розрахунки, необхідні описи (відповідно до завдання вище) та вихідний код програм.

Назва файлу: Прізвище_група

Роботи, виконані не за варіантом, не приймаються.

3.4 Контрольні питання

1. Що таке фіктивні змінні і для чого вони використовуються?
2. Яка різниця між опорним кодуванням та кодуванням з одним активним станом?
3. Чому важливо використовувати параметр drop_first=True у pd.get_dummies() для регресійних моделей?
4. Які типи категоріальних змінних існують, і як їх відрізняти? Наведіть приклади.
5. Як кодувати впорядковані категоріальні змінні (наприклад, «якість будинка») для машинного навчання?
6. Чому логістична регресія вважається методом класифікації, а не регресії?
7. Яке ключове припущення робить наївний класифікатор Баєса?
8. Чому ліниві учні (наприклад, K-найближчих сусідів) повільні під час прогнозування?
9. Яка головна мета крос-валідації у машинному навчанні?
10. Які проблеми виникають, якщо не кодувати категоріальні змінні перед навчанням моделі?

Практична робота №4 – A/B тест та статистична значущість

Мета роботи: проведення власного експерименту щодо оцінки різного дизайну веб-сторінок, закріплення навичок проведення A/B тесту та підтвердження статистичної значущості результатів експерименту.

Практична робота присвячена дизайну статистичних експериментів, проведенню A/B тесту, та доведенню, чи спостережуваний ефект є наслідком випадковості або є статистично значущим. Робота виконується із використанням мови програмування Python та бібліотек Pandas, SciPy, NumPy, Matplotlib. За бажанням студента, допускається виконання роботи мовою програмування R. Для розв'язання задачі практичної роботи за узгодженням з викладачем студент може запропонувати свій набір даних.

4.1 Теоретичні відомості

Планування експериментів є наріжним каменем практичної статистики з додатками фактично в усіх галузях дослідження. Мета полягає в тому, щоб спланувати експеримент, який підтвердить або відхилить гіпотезу. Дослідники даних стикаються з потребою проводити безперервні експерименти, особливо щодо користувацького інтерфейсу та товарного маркетингу. У цій лекції подано огляд традиційного планування експериментів та обговорено кілька поширених завдань у науці про дані. У ній також буде розглянуто кілька часто цитованих у статистичному виведенні понять і дано пояснення їхнього сенсу й актуальності (або відсутності такої) для науки про дані.

Щоразу згадка статистичної значущості, p-значень або перевірки на основі t-статистики відбувається, як правило, в контексті класичного «конвеєра» статистичного висновку. Цей процес починається з гіпотези («препарат А кращий за наявний стандартний препарат», «ціна А прибутковіша за наявну ціну В»).

Експеримент (це може бути A/B-тест) призначений для перевірки гіпотези, побудованої таким чином, щоб забезпечувати незаперечні результати. Дані збирають і аналізують, і далі роблять висновок. Термін «висновок» відображає намір застосувати експериментальні результати, які передбачають лімітований набір даних, до більшого процесу або популяції. Типовий конвеєр експерименту показано на рис. 4.1.

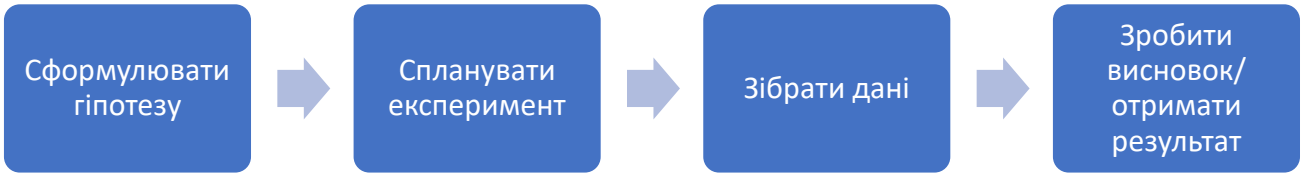

Рис. 4.1 – Класичний конвеєр експерименту.

4.1.1 Що таке A/B тестування?

A/B-тест – це експеримент із двома групами для визначення найкращого з двох варіантів, двох продуктів, двох процедур, двох лікарських засобів тощо. Нерідко один варіант із двох є стандартним існуючим варіантом або відсутній взагалі. Якщо використовується стандартний варіант (або ж він відсутній), то він називається контрольним. Типова гіпотеза полягає в тому, що пропонований варіант кращий за контрольний.

Приклади A/B тестування:

- тестування двох методів обробітку ґрунту, щоб визначити, яка з них призводить до найкращого проростання саджанців;
- тестування двох методів лікування, щоб визначити, який з них ефективніше пригнічує рак;
- тестування двох цін, щоб визначити, яка з них приносить більше чистого прибутку;
- тестування двох заголовків веб-сторінки, щоб визначити, який із них породжує більше натискань;
- тестування двох веб-оголошень, щоб визначити, яке з них генерує більше конверсій.

Під час A/B тесту необхідно чітко визначати, чи є отриманий ефект наслідком різних варіантів або випадковістю? Належний A/B-тест має випробовуваних, які можуть бути віднесені до того чи іншого варіанта експерименту. Піддослідним може бути людина, саджанець, відвідувач веб-сайту; головне, що піддослідному пропонується варіант експерименту. В ідеальному випадку випробовуваних рандомізують (призначають у випадковому порядку) за двома запропонованими варіантами. Завдяки цьому ви знаєте, що будь-яка різниця між тестовими групами відбувається внаслідок одного з двох:

- ефекту різних варіантів експерименту;
- чистої випадковості (тобто випадок, можливо, призвів до того, що результативніші випробовувані були природним чином сконцентровані в A або B).

4.1.2 Для чого необхідна перевірка значущості?

Перевірки гіпотез, так звані перевірки значущості, набули значного поширення в традиційному статистичному аналізі, що зустрічається в опублікованих дослідженнях. Такі перевірки призначені для того, щоб допомогти дізнатися, чи може випадковість бути відповідальною за спостережуваний ефект. У типовій ситуації A/B-тест конструюється з урахуванням гіпотези. Наприклад, гіпотеза може полягати в тому, що ціна B приносить вищий прибуток.

Навіщо нам потрібна гіпотеза? Чому не можна просто поглянути на результат експерименту і зупинитися на будь-якому варіанті, який працює краще? Відповідь криється у схильності людського розуму недооцінювати розмах природної випадкової поведінки. Один із проявів цієї схильності полягає в невмінні передбачати граничні події, або так званих «чорних лебедів» (концепція, згідно з якою важкопрогнозовані та рідкісні події, які мають значні наслідки, мають особливі характеристики). Ще одним її проявом є тенденція неправильно тлумачити випадкові події як такі, що мають ознаки певної значущості. Статистична перевірка гіпотез була винайдена як спосіб захистити дослідників від того, щоб бути обдуреним випадковістю.

Ключові терміни для перевірки гіпотез

- Нульова гіпотеза
    - Гіпотеза у тому, що виною всьому є випадковість.
- Альтернативна гіпотеза
    - Гіпотеза, що компенсує нульову (те, що ви сподіваєтесь довести).
- Одностороння перевірка
    - Перевірка гіпотези, коли кількість випадкових результатів підраховується лише в одному напрямі.
- Двостороння перевірка
    - Перевірка гіпотези, коли кількість випадкових результатів підраховується у двох напрямах.

Ключові терміни для статистичної значущості та p-значення

- p-значення
    - З урахуванням випадкової моделі, яка втілює нульову гіпотезу, p-значення є ймовірністю випадкового отримання результатів настільки ж незвичайних або граничних, як і результати спостереження.
- Альфа
    - Імовірнісний поріг «незвичайності», який випадкові результати повинні перевершити, щоб фактичні наслідки вважалися статистично значимими. Традиційно або 1%, або 5%.
    - *Синонім:* рівень значущості.
- Помилка 1-го роду

    - Помилковий висновок у тому, що ефект є реальним (тоді, як і зумовлений випадковістю).
- Помилка 2-го роду
    - Помилковий висновок у тому, що ефект зумовлений випадковістю (тоді, як і він є дійсним).
- Якщо результат лежить поза випадкової варіації, то прийнято казати, що він є статистично значущим.

4.2 Приклад розв'язування задачі

Давайте розберемо на прикладі, як визначити чи є отриманий результат статистично значущим, коли експеримент проводиться для двох випадків. Початкові дані представлено на в табл. 4.1.

Табл. 4.1. Зібрані дані про покупки.

Результат	**Ціна А**	**Ціна В**
Куплено	200	182
Не куплено	23 539	22 406

Розрахуємо конверсію для варіанта А:

$$\frac{200}{23539 + 200} \cdot 100\% = 0{,}8425\% \qquad (4.1)$$

та конверсія варіанта В:

$$\frac{182}{22406 + 182} \cdot 100\% = 0{,}8057\% \qquad (4.2)$$

Отже маємо, що:

1. Ціна А забезпечує майже на 5% кращу конверсію!
2. Але різниця конверсій складає лише 0,0368%

Чи може краща конверсія бути зумовленою випадковістю? Для відповіді на це запитання застосуємо перестановочний тест:

1. Помістити картки, позначені 1 і 0, у коробку: вона представлятиме передбачувану спільну інтенсивність конверсії 382 одиниць та 45 945 нулів: 0,008246 = 0,8246%.
2. Перетасувати та витягти повторну вибірку розміру 23 739 (число n таке саме, що й у ціни А), записати число одиниць.

3. Записати число одиниць в 22 588, що залишилися (число n таке ж, що і у ціни B).
4. Записати різницю у частині одиниць.
5. Повторити кроки 2-4.
6. Відповісти на запитання: як часто спостерігалася різниця ≥ 0,0368?

Ми можемо оцінити p-значення з нашого перестановочного тесту шляхом взяття частки числа разів, коли перестановочний тест породжує різницю, рівну чи більшу, ніж різниця, що спостерігається. p-значення становить 0,684, а значить, ми очікувано будемо досягати такого ж граничного результату, як і цей, або більш граничного через випадковість, що перевищує 68,4% часу. Вочевидь, отримане значення вище, ніж поріг $\alpha = 1\%$. **Результат не є статистично значущим!**

Проведемо тепер оцінку статистичної значущості за допомогою t-тесту. Функція ttest_ind в SciPy виконує t-тест Велча, який ідеально підходить для порівняння двох незалежних груп, коли ми не припускаємо рівних дисперсій – типовий сценарій у реальному житті. Ось як це працює:

1. Обчислимо спостережувану різницю між середніми значеннями груп (середнє значення варіанту B мінус середнє значення варіанту A).
2. Оцінимо стандартну похибку цієї різниці, об'єднавши індивідуальні дисперсії кожної групи (не припускаючи, що вони рівні).
3. Обчислимо t-статистику:

$$t = \frac{mean_B - mean_A}{\sqrt{\frac{s_B^2}{n_B} + \frac{s_A^2}{n_A}}}$$

   де s_A, s_B – стандартні відхилення вибірки, а n_A, n_B – розміри вибірки.
4. Визначимо p-значення, порівнявши цю t-статистику з t-розподілом (з поправкою на ступені свободи), що покаже нам ймовірність спостереження такої різниці, якщо групи були б дійсно ідентичними.

Для проведення t-тесту використаємо наступний код:

Код

```
from scipy.stats import ttest_ind

print(ttest_ind(A, B, equal_var=False))
```

Вивід

```
TtestResult(statistic=np.float64(0.43753959625470934),
pvalue=np.float64(0.6617221078137985), df=np.float64(46289.77348180466))
```

де масив A містить 200 одиниць та 23539 нулів, масив B – 182 одиниці та 22406.

p-значення вище обраного порогу $\alpha = 1\%$, а отже як і у випадку перестановочного тесту отримали, що результат не є статистично значущим.

4.3 Індивідуальне завдання

Вітаємо, Вас запросили працювати в компанію Google аналітиком даних і першим вашим завданням є провести тестування, який заголовок YouTube дає довші перегляди відео. Ваш колега є новатором в сфері дизайну веб-сторінок та запропонував замінити «класичну» іконку на нову (див. рис. 4.2. Знак оклику все змінює!). Він каже, що нова іконка дає на 10% більше переглядів та різниця є статистично значущою. Перевірте вашого колегу-новатора!

«Класична» іконка Новаторська іконка

Рис. 4.2 – Класична та новаторська іконки YouTube.

Для цього необхідно:

1. Завантажити дані індивідуального варіанту за посиланням на порталі дистанційної освіти.
2. Використовуючи Pandas відобразити перші 5 рядків даних власного варіанта.
3. Побудувати коробкові діаграми для даних обох варіантів, розрахувати середнє та медіану.
4. Самостійно реалізувати перестановочний тест в Python, використовуючи псевдокод, наведений нижче. Оцінити p-значення за допомогою перестановочного тесту та порівняти із альфа. Зробити висновок про статистичну значущість.
    a. Альфа взяти 1% для непарних варіантів, 5% для парних.
5. Використати t-test для оцінки значущості. Порівняти результати перестановочного A/B тесту і t-тесту.
6. Чи правильний висновок зробив ваш колега-новатор?
7. Навести висновок, який буде містити:
    a. Чи різниця між варіантами є статистично значущою із відповідними p-значеннями
    b. Якщо так, то який варіант краще та на скільки відсотків.

Псевдо-код перестановочного A/B тесту

1. Об'єднати результати з різних груп у єдиний набір даних.

2. Перетасувати об'єднані дані, потім у випадковому порядку витягти (без повернення) повторну вибірку того самого розміру, що й група А (очевидно, що вона міститиме небагато даних з іншої групи).
3. З даних, що залишилися, у випадковому порядку витягти (без повернення) повторну вибірку того ж розміру, що і група В.
4. Розрахувати середній час перегляду відео для повторних вибірок та записати; це буде однією ітерацією перестановки.
5. Повторити попередні кроки R разів (наприклад, 200) для отримання перестановного розподілу перевірочної статистики.

Тепер повернемося до різниці між групами і порівняємо її з набором перестановлених різниць:

- Якщо різниця, що спостерігається, переконливо лежить в межах набору перестановлених різниць, то ми нічого не довели – різниця знаходиться всередині діапазону того, що може спричинити випадковість.
- Однак, якщо різниця, що спостерігається лежить поза більшою частиною перестановочного розподілу, ми приходимо до висновку, що випадковість не несе відповідальності. Говорячи технічною мовою, різниця є *статистично значущою.*

Розрахувати p-значення як відношення кількості результатів із часом перегляду більше або рівного різниці між А та В до загальної кількості експериментів.

Робота має бути представлена у вигляді звіту (файл pdf). Звіт має містити титульний аркуш, номер варіанта, розрахунки, необхідні описи (відповідно до завдання вище) та вихідний код програм.

Назва файлу: Прізвище_група

Роботи, виконані не за варіантом, не приймаються.

4.4 Контрольні питання

1. Як перестановочний тест допомагає визначити статистичну значущість?
2. Яку роль відіграє альтернативна гіпотеза?
3. Що відрізняє односторонню від двосторонньої перевірки?
4. Чому досліднику не слід обирати метрику після експерименту?
5. Яка головна обмеженість надмірного звертання до p-значення?
6. Як перестановочний тест реалізує нульову гіпотезу?
7. Чому важливо мати попередньо визначену гіпотезу?
8. Який основний висновок про статистичну значущість та практичну корисність?
9. Яка основна мета дизайну експериментів в статистиці?
10. Чому важлива рандомізація в A/B-тестуванні?

Практична робота №5 – Багаторукі бандити і дизайнер-новатор. Проведення тесту для довільної кількості варіантів

Мета роботи: закріплення навичок проведення тесту для довільної кількості варіантів із використанням алгоритму є-жадібного бандиту, візуалізації та аналізу результатів тесту.

Практична робота присвячена дизайну експериментів з багатьма варіантами, алгоритму ϵ-жадібного бандита для множинного тестування та визначення кращого варіанта. Робота виконується із використанням мови програмування Python та бібліотек Pandas, SciPy, NumPy, Matplotlib. За бажанням студента, допускається виконання роботи мовою програмування R. Для розв'язання задачі практичної роботи за узгодженням з викладачем студент може запропонувати свій набір даних.

5.1 Теоретичні відомості

5.1.1 Хто такі багаторукі бандити?

Багаторукі бандити пропонують підхід до тестування, особливо веб-тестування, який дає змогу виконувати явну оптимізацію й ухвалювати швидші рішення, ніж традиційний статистичний підхід до планування експериментів.

Ключові терміни для багаторуких бандитів

- Багаторукий бандит
    - Уявний ігровий автомат з кількома важелями, або руками, на які гравець може натискати на вибір, при цьому кожна рука має різний виграш; тут узятий як аналогія багатоваріантного експерименту.
- Важіль (рука)
    - Варіант в експерименті (наприклад, «заголовок А у веб-тесті»).
- Виграш
    - Експериментальний аналог виграшу в автоматі (наприклад, «клієнт клацає на посилання»).

Багаторукі бандити пропонують підхід до тестування, особливо веб-тестування, який дозволяє здійснювати явну оптимізацію та швидше приймати рішення, ніж традиційний статистичний підхід до розробки експериментів.

Традиційний A/B-тест передбачає збір даних в експерименті відповідно до визначеного дизайну для відповіді на конкретне питання, наприклад: «Що краще, лікування А чи лікування В?». Передбачається, що після отримання відповіді на

це питання експеримент закінчується і ми переходимо до дій на основі отриманих результатів.

Виникає кілька труднощів у такому підході:

- по-перше, наша відповідь може бути непереконливою: «ефект не доведений». Іншими словами, результати експерименту можуть вказувати на ефект, але якщо ефект є, ми не маємо достатньо великої вибірки, щоб довести його (відповідно до традиційних статистичних стандартів). Яке рішення ми приймаємо в такому випадку?
- по-друге, ми можемо захотіти почати використовувати результати, отримані до завершення експерименту.
- по-третє, ми можемо захотіти мати право змінити свою думку або спробувати щось інше на основі додаткових даних, отриманих після завершення експерименту.

Традиційний підхід до експериментів і перевірки гіпотез датується 1920-ми роками і є досить негнучким. Поява потужних комп'ютерів і програмного забезпечення дозволила застосовувати більш гнучкі підходи. Більше того, наука про дані (і бізнес загалом) не так переймається статистичною значущістю, як оптимізацією загальних зусиль і результатів.

Алгоритми бандитів, які дуже популярні в веб-тестуванні, дозволяють тестувати кілька варіантів одночасно і доходити висновків швидше, ніж традиційні статистичні моделі. Вони отримали свою назву від ігрових автоматів, які використовуються в азартних іграх, також званих однорукими бандитами (оскільки вони налаштовані таким чином, що витягують гроші з гравця в постійному потоці). Якщо уявити ігровий автомат з більш ніж одним важелем, кожен з яких виплачує гроші з різною швидкістю, ви отримаєте багаторукого бандита, що є повною назвою цього алгоритму. Ваша мета – виграти якомога більше грошей і, більш конкретно, якомога швидше визначити і зупинитися на виграшній руці. Складність полягає в тому, що ви не знаєте, з якою загальною швидкістю виплачують руки – ви знаєте тільки результати окремих потягувань за руки.

Припустимо, що кожна «виграш» становить однакову суму, незалежно від того, яка рука. Відмінність полягає в ймовірності виграшу. Припустимо, що спочатку ви пробуєте кожну ручку 50 разів і отримуєте такі результати:

- ручка A: 10 виграшів із 50;
- ручка B: 2 виграші із 50;
- ручка C: 4 виграші із 50.

Один з крайніх підходів полягає в тому, щоб сказати: «Схоже, ручка A є переможцем – давайте припинимо пробувати інші ручки і залишимося при A». Це дозволяє повною мірою скористатися інформацією, отриманою під час початкового випробування. Якщо A дійсно кращий, ми отримуємо перевагу на ранньому етапі. З іншого боку, якщо B або C дійсно кращі, ми втрачаємо будь-яку можливість це виявити.

Інший крайній підхід полягає в тому, щоб сказати: «Все це виглядає як випадковість – давайте продовжувати вибирати їх порівну». Це дає максимальну можливість альтернативам A проявити себе. Однак у процесі цього ми застосовуємо те, що здається гіршими методами лікування. Як довго ми можемо це дозволяти?

Алгоритми епсілон-жадібних бандитів використовують гібридний підхід: ми починаємо частіше вибирати A, щоб скористатися його очевидною перевагою, але не відмовляємося від B і C. Ми просто вибираємо їх рідше. Якщо варіант A продовжує перевершувати інші, ми продовжуємо перерозподіляти ресурси (використання) від варіантів B і C і частіше використовуємо варіант A. Якщо, з іншого боку, варіант C починає показувати кращі результати, а варіант A – гірші, ми можемо перерозподілити використання з варіанту A назад на варіант C. Якщо один з них виявляється кращим за варіант A, а це було приховано в початковому випробуванні через випадковість, тепер він має можливість проявитися в подальших випробуваннях.

5.1.2 Застосування епсілон-жадібних бандитів до веб-тестування

Тепер подумайте про застосування цього до веб-тестування. Замість декількох ручок ігрових автоматів, ви можете мати декілька пропозицій, заголовків, кольорів тощо, які тестуються на веб-сайті. Клієнти або клацають (це «виграш» для продавця), або не клацають. Спочатку пропозиції показуються випадково і рівномірно. Однак, якщо одна пропозиція починає перевершувати інші, її можна показувати («витягувати») частіше. Але якими мають бути параметри алгоритму, що змінює частоту витягування? На які частоти витягування і коли ми повинні змінити?

Ось один простий алгоритм, алгоритм епсилон-жадібного бандита для двох варіантів:

Алгоритм ϵ-жадібного бандита

1. Згенеруйте рівномірно розподілене випадкове число від 0 до 1.
2. Якщо число лежить в діапазоні від 0 до епсилон (де епсилон – це число від 0 до 1, зазвичай досить мале), підкиньте чесну монету (ймовірність 50/50):
    a. Якщо випала решка, покажіть пропозицію A.
    b. Якщо випав орел, покажіть пропозицію B.
3. Якщо число ≥ епсилон, покажіть ту пропозицію, яка на даний момент має найвищий рівень відгуку.

Епсилон – це єдиний параметр, який керує цим алгоритмом. Якщо епсилон дорівнює 1, ми отримуємо стандартний простий експеримент A/B (випадковий розподіл між A і B для кожного суб'єкта). Якщо епсилон дорівнює 0, ми отримуємо суто жадібний алгоритм – такий, що вибирає найкращий доступний

варіант (локальний оптимум). Він не шукає подальших експериментів, а просто розподіляє суб'єктів (відвідувачів веб-сайту) за найкращим варіантом.

5.2 Приклад розв’язування задачі

Розглянемо набір даних натискання на рекламні оголошення для декількох варіантів реклами A, B, C:

clicks.csv

```
A,B,C
1,1,1
1,1,0
1,0,0
1,0,0
0,1,0
1,0,1
1,1,0
0,0,1
1,0,0
0,0,1
```

Алгоритм багаторукого бандита безпосередньо розширюється на випадок з багатьма варіантами.

Модифікація алгоритму для 3 варіантів:

1. Генеруйте випадкове число r від 0 до 1.
2. Якщо $r < \epsilon$:
   - Виберіть випадково один з варіантів A, B або C з однаковою ймовірністю (1/3 кожен).
3. Якщо $r \geq \epsilon$:
   - Покажіть пропозицію з найвищим історичним коефіцієнтом відгуку (коефіцієнт = кліки / покази).

Застосуємо цей алгоритм для декількох кроків на даних clicks.csv.

Зафіксуємо $\epsilon = 0.1$.

Крок 1

Випадкове r = 0,05 (< 0,1 → досліджувати).
Випадково вибрано B (варіант дослідження).
Показано B → Використано рядок 1: B клік = 1 (натиснуто).
Результат: B має 1 клік / 1 показ → коефіцієнт B = 1,0 (A: 0, C: 0).

Крок 2

Випадкове r = 0,15 (≥ 0,1 → використання).

Найвищий коефіцієнт = B (1,0) → Показано B.
Використано рядок 2: B клік = 1 (клацнуто).
Результат: B має 2 кліки / 2 покази → коефіцієнт B = 1,0 (A: 0, C: 0).

Крок 3
Випадкове r = 0,08 (< 0,1 → досліджувати).
Випадково вибрано C (вибір дослідження).
Показано C → Використано рядок 3: C клік = 1 (клацнуто).
Результат: C має 1 клік / 1 показ → коефіцієнт C = 1,0 (B: 1,0, A: 0).

Крок 4
Випадкове r = 0,20 (≥ 0,1 → використання).
Рівні коефіцієнти: B (1,0) = C (1,0) → Вибрано B (довільний вирішальний фактор).
Показано B → Використано рядок 4: B клік = 0 (не натиснуто).
Результат: B має 2 кліки / 3 покази → коефіцієнт B = 0,67 (C: 1,0, A: 0).

Крок 5
Випадкове r = 0,12 (≥ 0,1 → використання).
Найвищий показник = C (1,0) → Показано C.
Використано рядок 5: C клік = 0 (не натиснуто).
Результат: C має 1 клік / 2 покази → Показник C = 0,5 (B: 0,67, A: 0).

Крок 6
Випадкове r = 0,03 (< 0,1 → досліджувати).
Випадково вибрано A (вибір дослідження).
Показано A → Використано рядок 6: A клік = 1 (натиснуто).
Результат: A має 1 клік / 1 показ → коефіцієнт A = 1,0 (B: 0,67, C: 0,5).

Крок 7
Випадкове r = 0,15 (≥ 0,1 → використання).
Найвищий коефіцієнт = A (1,0) → Показано A.
Використано рядок 7: A клік = 1 (натиснуто).
Результат: A має 2 кліки / 2 покази → коефіцієнт A = 1,0 (B: 0,67, C: 0,5).

Дані експериментів узагальнено в таблиці 5.1, а результати – показано в табл. 5.2.

Табл. 5.1. Кроки алгоритму ϵ-жадібного бандита.

Крок	r	$r <$ 0.1?	Дія	Показано	Клік	Кліки (A/B/C)	Відгуки (A/B/C)
1	0.05	Так	Досл.	**B**	1 (B=1)	A:0, B:1, C:0	A:0, **B:1.0**, C:0
2	0.15	Ні	Викор.	**B**	1 (B=1)	A:0, B:2, C:0	A:0, **B:1.0**, C:0
3	0.08	Так	Досл.	**C**	1 (C=1)	A:0, B:2, C:1	A:0, B:1.0, **C:1.0**
4	0.20	Ні	Викор.	**B**	0 (B=0)	A:0, B:3, C:1	A:0, **B:0.67**, C:1.0
5	0.12	Ні	Викор.	**C**	0 (C=0)	A:0, B:3, C:2	A:0, B:0.67, **C:0.5**
6	0.03	Так	Досл.	**A**	1 (A=1)	A:1, B:3, C:2	**A:1.0**, B:0.67, C:0.5
7	0.15	Ні	Викор.	**A**	1 (A=1)	A:2, B:3, C:2	**A:1.0**, B:0.67, C:0.5

Алгоритм спочатку досліджує всі варіанти (кроки 1, 3, 6), щоб не пропустити варіанти з високою ефективністю (наприклад, A на кроці 6). Після накопичення даних він використовує найкращий варіант (A після кроку 6). Без дослідження він міг би ніколи не протестувати A (оскільки B/C домінували в ранніх даних).

Баланс між дослідженням і використанням: 10% дослідження (ε=0,1) гарантує, що ми навчаємося, одночасно максимізуючи продуктивність.

Табл. 5.2. Результати експерименту.

Пропозиція	Показано	Кліки	Відгуки
A	2	2	100%
B	3	2	67%
C	2	1	50%

За наявними даними бачимо, що варіант A є кращим. Вочевидь, на практиці 7 експериментів недостатньо. Процедуру треба повторити кілька сотень разів, щоби отримати результат.

5.3 Індивідуальне завдання

Повний натхнення від першого проєкту YouTube*!* та вашої допомоги дизайнер-новатор вирішив модернізувати іконки для всієї лінійки продуктів від Google! Він був в захваті, коли дізнався, що ви знайомі із багаторукими та ще й жадібними бандитами! Тепер він сповнений ідей та хоче вирішити, який варіант

є кращим серед 4-6 видів нових іконок для Gmail! Тепер він каже, що при використанні варіанту іконки №3 користувачі надсилають на 15% більше листів Gmail. Перевірте це!

Для цього необхідно:

1. Завантажити дані індивідуального варіанту за посиланням на порталі дистанційної освіти, використовуючи Pandas відобразити перші 5 рядків.
2. Запрограмувати алгоритм ϵ-жадібного бандита. Встановити $\epsilon = 0{,}1$.
3. При проведені тесту із ϵ-жадібним бандитом всі дані не є доступними відразу, натомість ви показуєте користувачам різні сторінки та збираєте дані в режимі онлайн. Використовуючи запрограмований алгоритм ϵ-жадібного бандита, зібрати як мінімум 300 прикладів. Приклади брати по одному з даних за індивідуальним варіантом.
4. Показати:
    a. Діаграму, де зобразити скільки разів було показано кожну із сторінок і (на ній же) кількість відправлених листів для кожної із сторінок
    b. Графік відношення відправлених листів до показів сторінки (для кожної із сторінок) із часом. Для зображення використовувати сукупні дані зібрані алгоритмом, за час взяти ітерацію.
5. Зазначити, який варіант є кращим.

1. Згенерувати рівномірно розподілене випадкове число в інтервалі між 0 та 1.
2. Якщо число знаходиться між 0 та ϵ (де ϵ (епсілон) — це число між 0 та 1 у типовій ситуації досить мале), випадково обрати один з варіантів іконки Gmail (№1, 2, 3, 4…).
3. Якщо число більше або дорівнює ϵ, то показати будь-яку пропозицію, яка дотепер мала найвище відношення відправлених листів.

Робота має бути представлена у вигляді звіту (файл pdf). Звіт має містити титульний аркуш, номер варіанта, розрахунки, необхідні описи (відповідно до завдання вище) та вихідний код програм.

Назва файлу: Прізвище_група

Роботи, виконані не за варіантом, не приймаються.

5.4 Контрольні питання

1. Яка головна перевага алгоритму епсілон-жадібних бандитів над традиційним A/B-тестуванням у веб-експериментах?
2. Що контролює параметр ϵ в алгоритмі епсілон-жадібних бандитів?

3. Коли згенероване випадкове число менше за ϵ, які дії виконує алгоритм?
4. Як алгоритм епсілон-жадібних бандитів балансує між дослідженням і використанням?
5. Якщо $\epsilon = 0$, до якого типу алгоритму зводиться система? Якщо $\epsilon = 1$, як поводиться алгоритм?
6. У прикладі clicks.csv, який варіант буде мати найвищий коефіцієнт кліків після 10 кроків?
7. Що сталося б, якби епсилон в прикладі було встановлено на 0,5 замість 0,1?
8. Чи достатньо 7 прикладів для визначення кращого варіанту за допомогою епсілон-жадібних бандитів?
9. Яка мета етапів дослідження та використання в алгоритмі?
10. Порівняйте A/B тестування та алгоритм епсілон-жадібних бандитів для випадку 2 варіантів. Які переваги та недоліки кожного з алгоритмів можете навести?

Оцінювання результатів навчання

Оцінювання практичних робіт

Кожна робота оцінюється за 100-бальною шкалою. Оцінювання відбувається строго у відповідності до пунктів індивідуального завдання, наведених в кожній практичній роботі, та пропорційно до ступеня повноти та коректності виконання кожного з них. Важливими вимогами до звіту з практичної роботи є:

1. Чіткий структурований опис проведеного дослідження, наведення необхідних графічних ілюстрацій.
2. Аналіз результатів, власні припущення та висновки студента, щодо отриманих закономірностей.
3. Кожна робота має обов'язково містити лістинг програмного коду, коректність якого та відповідність завданню оцінюється викладачем, програмний код студент має розуміти та виконати самостійно.
4. Роботи, виконані не за варіантом, не приймаються.

Вимоги до звіту

Звіт з виконання практичної роботи має містити:

- обкладинку в корпоративному стилі НТУ «Дніпровська політехніка» із зазначенням групи, ПІБ студента, викладача та теми роботи;
- номер варіанта, мета роботи, повний текст завдання;
- за кожним пунктом завдання опис, що було виконано, які результати отримано. Результати кожного пункту мають бути проаналізовані здобувачем;
- в кінці роботи має бути обов'язково наявний лістинг коду.

Звіт розміщується на сайті дистанційної освіти у відповідному розділі із завданням і оцінюється виходячи зі 100 балів.

Рекомендовані джерела інформації

1. Математичні методи інтелектуального аналізу даних: [навчальний посібник для здобувачів першого рівня вищої освіти спеціальності 124 Системний аналіз] / Т. Шабельник, О. Дяченко. – Маріуполь: МДУ, 2021. – 163 с
2. Кононова К. Ю. Машинне навчання: методи та моделі / К. Ю. Кононова. – Харків: ХНУ імені В. Н. Каразіна, 2020. – 301 с.
3. Хабарлак К.С. Самонавчання складних систем [Електронний ресурс] : конспект лекцій для здобувачів ступеня магістра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / К.С. Хабарлак, Т.А. Желдак ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 112 с.
4. Документація бібліотеки машинного навчання scikit-learn. URL: https://scikit-learn.org .
5. Документація бібліотеки аналізу даних в Python: pandas. URL: https://pandas.pydata.org/ .
6. Practical Statistics for Data Scientists / P. Bruce, A. Bruce, P. Gedeck. – O'Reilly Media, 2020.
7. Хабарлак К.С. Аналіз та обробка великих даних [Електронний ресурс] : конспект лекцій для здобувачів ступеня магістра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / К.С. Хабарлак, Т.В. Хом’як ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 111 с.

Навчальне видання

Хабарлак Костянтин Сергійович
Коряшкіна Лариса Сергіївна
Желдак Тимур Анатолійович
Хом'як Тетяна Валеріївна

АНАЛІЗ ДАНИХ ТА ЗНАНЬ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
зі спеціальності 124 Системний аналіз
(F4 Системний аналіз та наука про дані)

Видано в авторській редакції.

Електронний ресурс.
Підписано до видання 12.06.2025. Авт. арк. 4,5.

Національний технічний університет «Дніпровська політехніка».
49005, м. Дніпро, просп. Дмитра Яворницького, 19.