

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет будівництва і архітектури

МОДЕЛЮВАННЯ ЗАДАЧ ШТУЧНОГО ІНТЕЛЕКТУ

Методичні вказівки
до виконання розрахунково-графічної роботи
для підготовки здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальностями 122 «Комп'ютерні науки»,
126 «Інформаційні системи і технології»

Київ 2024

УДК 004.8

М74

Укладачі: Т. А. Гончаренко, канд. техн. наук, доцент;

В. В. Гоц, канд. техн. наук, доцент;

С. Ю. Долгополов, асистент

Рецензент І. А. Ачкасов, д-р техн. наук, професор

Відповідальний за випуск Т. А. Гончаренко, канд. техн. наук,
доцент

*Затверджено на засіданні кафедри інформаційних технологій,
протокол № 9 від 27 березня 2024 року.*

В авторській редакції.

Моделювання задач штучного інтелекту : методичні вказівки
М74 до виконання розрахунково-графічної роботи / уклад. :
Т. А. Гончаренко, В. В. Гоц, С. Ю. Долгополов. – Київ : КНУБА,
2024. – 34 с.

Містять зміст, порядок оформлення і вказівки до виконання
розрахунково-графічної роботи бакалавра.

Призначено для здобувачів першого (бакалаврського) рівня
вищої освіти за спеціальностями 122 «Комп'ютерні науки»,
126 «Інформаційні системи і технології».

© КНУБА, 2024

Зміст

Загальні положення	4
1. Структура розрахунково-графічної роботи	5
2. Оформлення пояснювальної записки	6
3. Завдання для розрахунково-графічних робіт	9
4. Порядок оцінювання розрахунково-графічних робіт	11
5. Зразок виконання практичної частини дослідження	12
Перелік рекомендованої літератури	31

Загальні положення

Розрахунково-графічна робота є підсумковим етапом вивчення здобувачами вищої освіти спеціальностей 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології» дисципліни «Моделювання задач штучного інтелекту».

Під час виконання розрахунково-графічної роботи здобувач вищої освіти повинен продемонструвати вміння самостійно працювати і застосовувати на практиці теоретичні знання, отримані під час вивчення дисципліни.

Метою розрахунково-графічної роботи є:

- розвиток аналітичного мислення та навичок критичного аналізу даних через реалізацію конкретних проєктів з використанням методів штучного інтелекту;
- закріплення та поглиблення знань з основних тем і концепцій, вивчених під час курсу, шляхом їх практичного застосування в процесі виконання проєктів;
- систематизація отриманих знань та навичок з моделювання задач штучного інтелекту для підготовки здобувачів до вирішення реальних задач у професійній діяльності;
- розвиток навичок самостійної роботи з науковою та технічною літературою, а також з використанням сучасних програмних інструментів для аналізу даних і розробки моделей штучного інтелекту;
- формування вмінь планувати дослідження, виконувати обчислювальні експерименти, аналізувати отримані результати та представляти їх у вигляді звітів і презентацій;
- залучення студентів до науково-дослідної роботи в області штучного інтелекту, стимулювання інтересу до подальшого розвитку в цьому напрямку.

Тематика розрахунково-графічних робіт обумовлена основними розділами дисципліни «Моделювання задач штучного інтелекту» – Data Science, Machine Learning, Deep Learning і аналіз даних, побудову та оптимізацію моделей машинного навчання, роботу з нейронними мережами, класифікацію текстів, розпізнавання образів, прогнозування часових рядів та розробку систем рекомендацій.

Розрахунково-графічна робота містить детально описане завдання з точно визначеними вихідними параметрами та список рекомендованої літератури за темою роботи. Здобувачам вищої освіти надається можливість обрати власну тему для розрахунково-графічної роботи, що може бути пов'язаною з їхнім науковим дослідженням або професійною діяльністю.

1. Структура розрахунково-графічної роботи

Розрахунково-графічна робота складається з наступних розділів:

- вступ;
- аналіз актуальних досліджень та розробок у сфері моделювання задач штучного інтелекту за тематикою;
- збір та підготовка даних: методи збору, очищення та аналізу даних;
- моделювання: розробка та тренування моделі/моделей машинного та/або глибокого навчання;
- оцінка ефективності моделі: методи валідації та аналіз показників якості моделі/моделей;
- висновки: аналіз отриманих результатів, їхнє значення та можливі напрями подальших досліджень;
- список використаних джерел;
- додатки: лістинги програм, масивні графічні матеріали тощо (за необхідності).

Обсяг звіту з розрахунково-графічної роботи, незалежно від теми, повинен бути не менше 20-ти сторінок.

Розділи розрахунково-графічної роботи мають наступний зміст:

1. Вступ (2–3 сторінки).

У вступі описується актуальність обраної теми, визначаються мета та завдання роботи, а також її наукова новизна та практичне значення.

Об’єкт дослідження – це процес або явище, що породжує проблемну ситуацію та обирається для дослідження. Об’єкт дослідження зазначається через назву певної системи (обладнання, пристрою, процесу, технології, програмного продукту, інформаційної технології, інтелектуального твору, явища тощо). Об’єкт дослідження має належати до класу узагальненого об’єкта діяльності фахівця обраної відповідної спеціальності.

Предмет дослідження – це те, що міститься у межах об’єкта й чому приділено основну увагу в дослідженні. Предмет дослідження зазначається у вигляді системи властивостей, характеристик, функцій об’єкта дослідження, на які безпосередньо має бути спрямовано дослідження (із зазначенням певних обмежень). Визначення предмета дослідження практично є конкретизацією наукової проблеми, що впливає із завдань дослідження.

2. Аналіз актуальних досліджень та розробок у сфері моделювання задач штучного інтелекту за тематикою (4–6 сторінок).

Огляд літератури та сучасних досліджень, що висвітлюють стан проблеми, яка досліджується. Аналізується історичний розвиток питання, сучасні підходи, методи та технології, що застосовуються в даній галузі.

3. Збір та підготовка даних (4–6 сторінок).

Описується процес збору даних, їхні джерела, методи очищення та аналізу. Включає опис використаних інструментів та алгоритмів для підготовки даних до подальшого моделювання, зокрема обробка пропущених значень, кодування категоріальних змінних, нормалізація числових даних тощо.

4. Моделювання (4–6 сторінок).

Розробка та тренування моделі або моделей машинного та/або глибокого навчання за вибраною тематикою. Описується вибір типу моделі, її архітектура, гіперпараметри та процес тренування. Описуються експерименти, проведені для вибору найкращої моделі.

5. Оцінка ефективності моделі (4–6 сторінок).

Методи валідації та аналіз показників якості моделі або моделей. Включає опис використаних метрик для оцінки точності, повноти, F1-скорю, ROC-кривої тощо. Аналізується, як модель впоралася з задачею, виявляються її сильні та слабкі сторони.

6. Висновки (1–2 сторінки).

Підсумок виконаної роботи, де аналізуються отримані результати у контексті визначених на початку цілей та завдань. Вказуються основні висновки, внесок роботи у розвиток обраної тематики, а також можливі напрями для подальших досліджень.

7. Список використаних джерел.

Перелік усієї наукової, методичної літератури та інших джерел, які були використані під час підготовки розрахунково-графічної роботи. Джерела оформляються згідно зі стандартом ДСТУ 8302:2015.

8. Додатки.

Розділ містить додаткові матеріали, які є релевантними для розуміння та оцінки виконаної роботи, але їх включення в основний текст було б недоцільним. Це можуть бути лістинги програм, масивні таблиці, графіки, схеми тощо.

2. Оформлення пояснювальної записки

Титульний аркуш пояснювальної записки оформляється за зразком, наведеним у додатку. Усі аркуші записки повинні бути пронумеровані. Першим аркушем є підписане керівником завдання на курсову роботу.

Текст роботи друкують з одного боку аркуша, на білому папері формату А4 (210x297 мм), залишаючи поля таких розмірів: ліве – не менш ніж 20 – 25 мм, праве – не менш ніж 10 мм, верхнє – не менш ніж 20 мм, нижнє – не менш ніж 20 мм. Варто використовувати шрифт Times New Roman 14 розміру з міжрядковим інтервалом 1,5.

1. *Рисунки та ілюстрації.*

Фотографії, креслення, схеми, графіки, карти і таблиці (ілюстрації) мають бути безпосередньо після тексту, де про них згадано вперше, або на наступній сторінці. Ілюстрації і таблиці, які розміщено на окремих сторінках роботи, включають до загальної нумерації сторінок. Ілюстрації позначають словом «Рис.» і нумерують послідовно у межах розділу, за винятком ілюстрацій, поданих у додатках.

Номер ілюстрації має складатися з номера розділу і порядкового номера ілюстрації, між якими ставлять крапку. Наприклад, «Рис. 3.2» (другий рисунок третього розділу). Номер ілюстрації, її назву і пояснювальні підписи розміщують послідовно під ілюстрацією. Якщо у роботі подано лише одну ілюстрацію, то її нумерують за загальними правилами.

2. *Таблиці.*

Таблиці нумерують арабськими цифрами у межах розділу, за винятком таблиць, що їх наведено у додатках.

Номер таблиці складається з номера розділу і порядкового номера таблиці, відокремлених крапкою, наприклад, табл. 2.1 (перша таблиця другого розділу). Таблиця може мати назву, яку друкують малими літерами (крім заголовної) і розміщують над таблицею. Назва має бути стислою і відбивати зміст таблиці.

Якщо таблицю неможливо помістити на одній сторінці, то її переносять на наступну з зазначенням номера таблиці, наприклад: Закінчення табл. 1. На наступній сторінці вказують тільки нумерацію граф без елементів головки таблиці, тобто у правому куті пишуть, наприклад, Закінчення табл. 4.

3. *Формули.*

Формули та рівняння розташовують безпосередньо після тексту, в якому про них згадують, посередині сторінки. Формули і рівняння у пояснювальній записці (за винятком формул і рівнянь, наведених у додатках) нумерують у межах розділу.

Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, відокремлених крапкою, наприклад, формула (1.3) – третя формула першого розділу. Номер формули або рівняння зазначають на рівні формули або рівняння в дужках у крайньому правому положенні на рядку.

Пояснення значень символів і числових коефіцієнтів формул чи рівнянь наводять безпосередньо під формулою у тій послідовності, в якій їх наведено у формулі чи рівнянні. Пояснення значень символів і числового коефіцієнта слід давати з нового рядка. Перший рядок пояснення починають з абзацу словом «де» без двокрапки.

Приклад: «Відомо, що

$$y = (model.predict(X_{test}) > 0.5).astype('float32'), \quad (1.3)$$

де y – вектор прогнозованих результатів;

$model.predict(X_{test})$ – вектор числових значень, переданих моделі для прогнозування;

0.5 – поріг, що використовується для визначення, чи вважається прогноз позитивним;

$.astype('float32')$ – метод, який конвертує булеві значення у значення з плаваючою точкою.

Список використаних літературних джерел

У списку наводять усі інформаційні джерела, які використовувались в процесі виконання роботи: галузеві керівні документи, стандарти, монографії, підручники, довідники, статті тощо. Джерела розміщуються у списку в порядку появи посилання на них в тексті записки і послідовно нумеруються арабськими цифрами. Допускається розміщувати літературні джерела за алфавітом. При цьому спочатку подаються літературні джерела, написані українською мовою, а далі англійською. До списку використаної літератури рекомендується включати такі джерела: основні нормативні документи і матеріали (державні й урядові); друковані джерела суспільно-політичного, соціального, економічного, природничо-наукового, соціально-культурного характеру; книги; статті; дисертації; автореферати; патентні документи; нормативно-технічні документи; каталоги промислового устаткування виробів; депоновані рукописи. Для виконання розрахунково-графічної роботи потрібно використати не менше 5 джерел. Назви літературних джерел наводять мовою, якою вони написані, – українською, англійською та іншими. Відомості про літературні джерела подають відповідно до вимог ДСТУ 8302:2015.

Презентація

До захисту розрахунково-графічної роботи готується презентація для доповіді-представлення роботи (3–5 хвилин). Оптимальним обсягом презентації для зазначеної тривалості вважається не більше 12 традиційних слайдів.

3. Завдання для розрахунково-графічних робіт

Задачі для курсової роботи обираються згідно з наведеними варіантами:

1. Прогнозування цін на житло за допомогою регресійних моделей.
2. Аналіз емоцій у текстах з використанням моделей NLP.
3. Розпізнавання об'єктів на зображеннях з використанням CNN.
4. Прогнозування погоди за даними часових рядів.
5. Автоматичне генерування тексту з використанням RNN.
6. Класифікація музичних жанрів за аудіофайлами.
7. Визначення фейкових новин за допомогою машинного навчання.
8. Автоматизований аналіз настроїв у соціальних мережах.
9. Розпізнавання рукописного введення з використанням глибокого навчання.
10. Оптимізація логістичних маршрутів за допомогою алгоритмів машинного навчання.
11. Прогнозування акційних цін за допомогою LSTM мереж.
12. Виявлення шахрайства з кредитними картками.
13. Рекомендаційна система для електронної комерції.
14. Класифікація зображень рентгенівських знімків.
15. Аналіз та прогнозування виборчих трендів.
16. Розпізнавання мови за аудіозаписами.
17. Виявлення дефектів на виробничих лініях за допомогою комп'ютерного зору.
18. Сегментація клієнтської бази за допомогою алгоритмів кластеризації.
19. Класифікації наукових статей.
20. Визначення віку людини за фотографією.
21. Аналіз ринкових трендів за даними соціальних медіа.
22. Оцінка кредитоспроможності за допомогою машинного навчання.
23. Визначення типів забруднення повітря.
24. Прогнозування врожайності сільськогосподарських культур.
25. Моделювання ризиків стихійних лих за даними сенсорів.
26. Виявлення плагіату в текстах.
27. Визначення оптимальних параметрів для виробництва продукції.
28. Прогнозування динаміки розвитку епідемій.
29. Аналіз і оптимізація енергоспоживання будівель.
30. Розробка системи раннього попередження про землетруси.
31. Класифікація видів тканин за допомогою комп'ютерного зору.
32. Виявлення аномалій у фінансових транзакціях.
33. Розробка персоналізованої медичної діагностичної системи.
34. Розпізнавання дорожніх знаків для автономних транспортних засобів.
35. Прогнозування результатів спортивних змагань.

36. Оптимізація складських запасів за допомогою прогнозування попиту.
37. Аналіз звукових сигналів для виявлення дефектів у промисловому обладнанні.
38. Розробка системи відстеження «дикої природи» за допомогою дронів.
39. Автоматизована система виявлення порушень на робочому місці.
40. Моніторинг якості води за допомогою машинного навчання.
41. Розробка системи безпеки на основі розпізнавання осіб.
42. Аналіз та прогнозування фондового ринку.
43. Виявлення та класифікація електронних спам-повідомлень.
44. Розробка системи допомоги водіям у навігації та уникненні ДТП.
45. Автоматизація процесу перевірки якості продукції на виробництві.
46. Аналіз та оптимізація трафіку в міській інфраструктурі.
47. Розробка системи ідентифікації музичних композицій.
48. Виявлення та класифікація біологічних об'єктів у мікроскопічних зображеннях.
49. Розробка системи підтримки прийняття рішень для аграрного сектора.
50. Аналіз соціальних мереж для виявлення трендів та впливових осіб.
51. Розробка системи для процесу збору даних з IoT-пристроїв.
52. Прогнозування зносу та терміну служби промислових деталей.
53. Розробка алгоритмів для глибокого аналізу відео.
54. Оптимізація маршрутів доставки за допомогою геопросторових даних.
55. Аналіз ефективності маркетингових кампаній.
56. Розробка інтерактивних освітніх платформ з адаптацією до користувача.
57. Аналіз емоційних станів користувачів на основі аналізу текстів у соціальних мережах.
58. Використання машинного навчання для автоматичного створення музики.
59. Розробка індивідуальних фітнес-планів.
60. Розробка рекомендаційної системи для онлайн-шопінгу.
61. Розробка інтелектуальної системи управління домашнім освітленням.
62. Використання нейронних мереж для розробки інтелектуальних ігрових стратегій.
63. Використання машинного навчання для оптимізації процесів в енергетиці.
64. Розробка системи розпізнавання та класифікації рослин та тварин.

Здобувач вищої освіти має можливість узгодити іншу тему, поза заданим списком, якщо вона відповідає напряму дисципліни та науковому або професійному спрямуванню здобувача.

4. Порядок оцінювання розрахунково-графічних робіт

1. Загальні положення

Оцінювання розрахункової роботи відбуватиметься на основі аналізу наступних факторів:

- Якість виконання та повнота роботи. Оцінюється, наскільки детально та вичерпно здобувач розробив кожен з розділів роботи, відповідність виконання поставленим завданням.

- Глибина аналізу та обґрунтування вибору методів. Враховується, наскільки обґрунтовано студент обирає методи збору даних, моделювання, аналізу результатів.

- Точність розрахунків та використання алгоритмів. Перевіряється правильність математичних розрахунків, використання алгоритмів та програмного коду.

- Якість оформлення. Враховуються загальний вигляд роботи, правильність оформлення джерел, дотримання стандартів оформлення тексту, таблиць, графіків та лістингів програм.

- Оригінальність та самостійність роботи. Оцінюється ступінь новизни та самостійності виконання роботи, вміння студента вносити власні пропозиції, ідеї та рішення.

- Захист роботи. Оцінюється вміння студента аргументовано представити свою роботу, відповідати на поставлені запитання, виявляти розуміння основних аспектів дослідження.

Оцінки (до 30 балів. Трансформуються в національну шкалу):

- **«Відмінно»**, повне правильне виконання роботи відповідно до зазначених вимог та критеріїв – 28–30 балів.

- **«Дуже добре»**, повне виконання роботи відповідно до зазначених вимог, незначні помилки у розрахунках – 23–27 балів.

- **«Добре»**, повне виконання роботи, незначні помилки у розрахункових формулах та/чи у розрахунках, певна невідповідність зазначеним вимогам та критеріям – 18–22 балів.

- **«Задовільно»**, повне виконання роботи, помилки у розрахункових формулах та/чи у розрахунках, певна невідповідність зазначеним вимогам та критеріям – 12–16 балів.

- **«Достатньо»**, неповне та/чи неправильне виконання роботи, суттєва невідповідність зазначеним вимогам та критеріям – 6–11 балів.

- **«Незадовільно»**, неповне та/чи неправильне виконання роботи, невідповідність зазначеним вимогам та критеріям – 1–5 балів.

- **«Роботу не зараховано»**, робота не виконана – 0 балів.

Шкала оцінювання: національна та ECST

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проєкту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

5. Зразок виконання практичної частини дослідження



Титанік – британський пасажирський лайнер компанії White Star Line, який затонув у північній частині Атлантичного океану вранці 15 квітня 1912 року після зіткнення з айсбергом під час свого першого рейсу з Саутгемптона до Нью-Йорка з приблизно 2224 пасажирами і членами екіпажу, які перебували на борту, понад 1500 загинули, що робить цю катастрофу однією з найсмертоносніших комерційних морських катастроф мирного

часу в сучасній історії. Хоча на борту було близько 2224 пасажирів і члена екіпажу, ми маємо дані про 1300 пасажирів. З цих 1300, близько 900 даних використовуються з навчальною метою, а решта 400 – з тестовою. У цьому завданні вам надано близько 400 тестових даних з відсутнім стовпчиком виживших. Вам треба змодельовати модель нейронної мережі, щоб передбачити, чи вижили пасажирів у тестових даних, чи ні. Як навчальні, так і тестові дані не є чистими (містять багато пропущених значень). Побудуйте модель з найкращою точністю.

Характеристика даних:

- Survival (виживання): 0 – ні, 1 – так;
- Pclass (номер класу): 1-й – вищий, 2-й – середній, 3-й – нижчий;
- sibsp (кількість братів і сестер / подружжя на борту): Sibling – брат, сестра, зведений брат, зведена сестра, Spouse – чоловік, дружина (коханки та наречені не враховуються);
- parch (кількість батьків / дітей на борту): Parent – мати, батько, Child – донька, син, падчерка, пасинок. Деякі діти подорожували лише з нянею, тому для них parch = 0;
- Ticket (квиток): номер квитка;
- Fare (вартість): пасажирський тариф;
- Cabin (каюта): номер каюти;
- Port of Embarkation (порт посадки): C – Cherbourg, Q – Queenstown, S – Southampton;
- Name (ім'я), Sex (стать), Age (вік).

Збережемо та імпортуємо набори даних (рис. 1.1 – рис. 1.2).

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen H	male		22	1	0	A/5 21171	7.25	S
3	2	1	1	Cummings, Mrs. John I	female		38	1	0	PC 17599	71.2833	C85
4	3	1	3	Heikinen, Miss. Lain	female		26	0	0	STON/O2. 3101282	7.925	S
5	4	1	1	Futrelle, Mrs. Jacques	female		35	1	0	113803	53.1	C123
6	5	0	3	Allen, Mr. William H	male		35	0	0	373450	8.05	S
7	6	0	3	Moran, Mr. James	male			0	0	330877	8.4583	Q
8	7	0	1	McCarthy, Mr. Timot	male		54	0	0	17463	51.8625	E46

Рис. 1.1. Датасет на Google Drive

Встановлення необхідних бібліотек та модулів:

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import tensorflow as tf
import keras
from keras.layers import Dense, Dropout, Input
from keras.models import Sequential
```

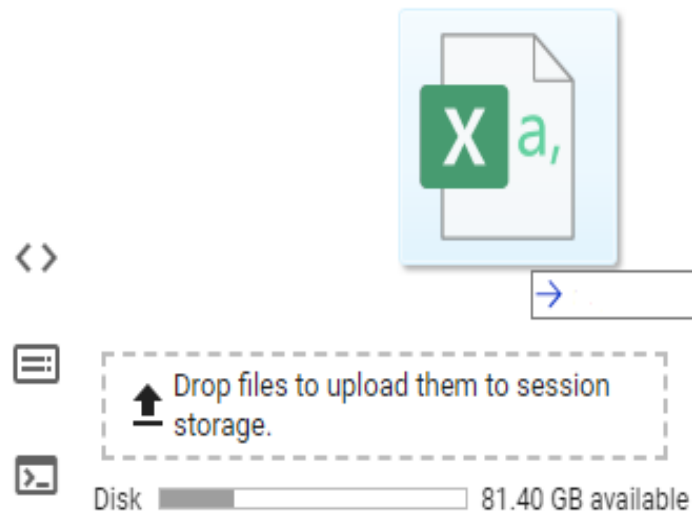


Рис. 1.2. Завантаження датасету до Google Colab

Завантаження наборів даних (рис. 1.3–1.4).

```
train_data = pd.read_csv('/content/train.csv')
test_data = pd.read_csv('/content/test.csv')
```

```
train_data.head()
test_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Рис. 1.3. Перші 5 рядків навчального набору даних

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

Рис. 1.4. Перші 5 рядків тестового набору даних

Ми бачимо, що загальна кількість стовпців у тестовому наборі даних (рис. 1.5) на один менше, ніж у навчальному. Відсутній стовпець у тестових даних – це Survived (Пасажири, що вижили) стовпець, який ми повинні передбачити.

```
print("Загальна кількість рядків у навчальному наборі даних ",
train_data.shape[0])
print("Загальна кількість стовпців у навчальному наборі даних ",
train_data.shape[1])
print("Загальна кількість рядків у тестовому наборі даних ",
test_data.shape[0])
print("Загальна кількість стовпців у тестовому наборі даних ",
test_data.shape[1])
```

```
Загальна кількість рядків у навчальному наборі даних 891
Загальна кількість стовпців у навчальному наборі даних 12
Загальна кількість рядків у тестовому наборі даних 418
Загальна кількість стовпців у тестовому наборі даних 11
```

Рис. 1.5. Загальна кількість рядків та стовпців у тестовому та навчальному наборах даних

Візуалізація та аналіз даних. Виконаємо візуалізацію кількості нульових значень в обох наборах даних (рис. 1.6–1.7).

```
plt.figure(figsize = (13,5))
plt.bar(train_data.columns, train_data.isna().sum())
plt.xlabel("Назва стовпця")
plt.ylabel("Кількість NULL значень у навчальному наборі даних")
```

```
plt.show()
```

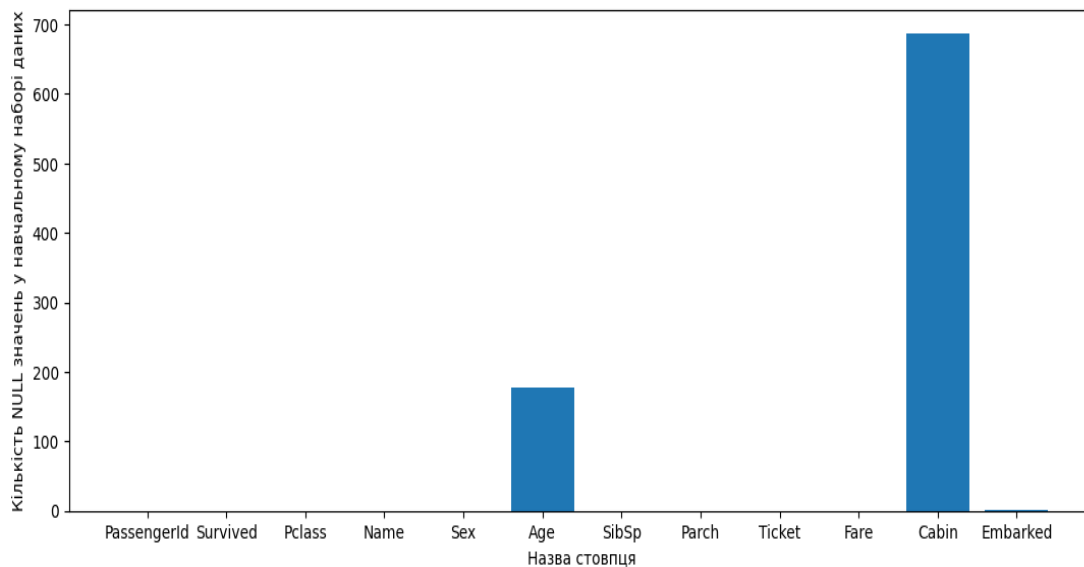


Рис. 1.6. Графік «Кількість NULL значень у навчальному наборі даних»

```
plt.figure(figsize = (13,5))  
plt.bar(test_data.columns, test_data.isnull().sum().values, color = 'red')  
plt.xlabel("Назва стовпця")  
plt.ylabel("Кількість NULL значень у тестовому наборі даних")  
plt.show()
```

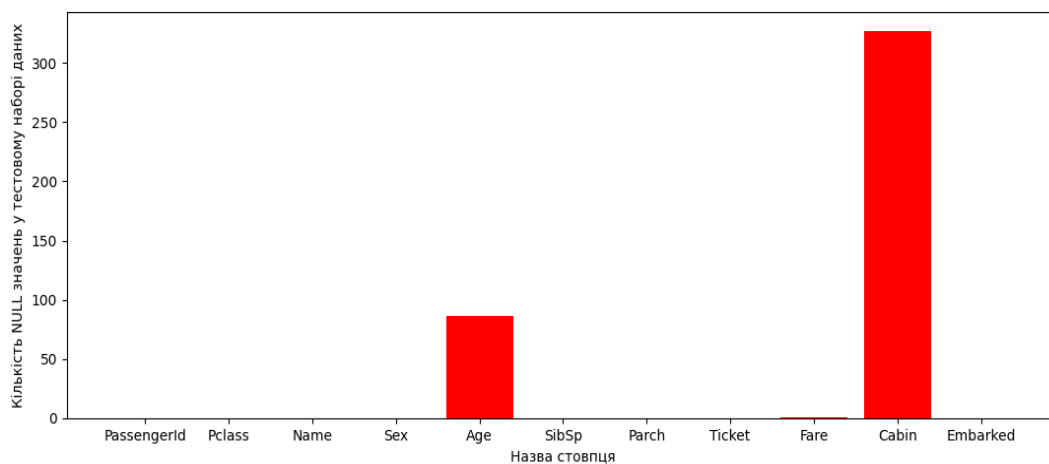


Рис. 1.7. Графік «Кількість NULL значень у тестовому наборі даних»

Виконаємо візуалізацію кількості пасажирів, що вижили (рис. 1.8).

```
# Візуалізація кількості пасажирів, що вижили  
sns.countplot(x='Survived', data = train_data)  
plt.show()
```

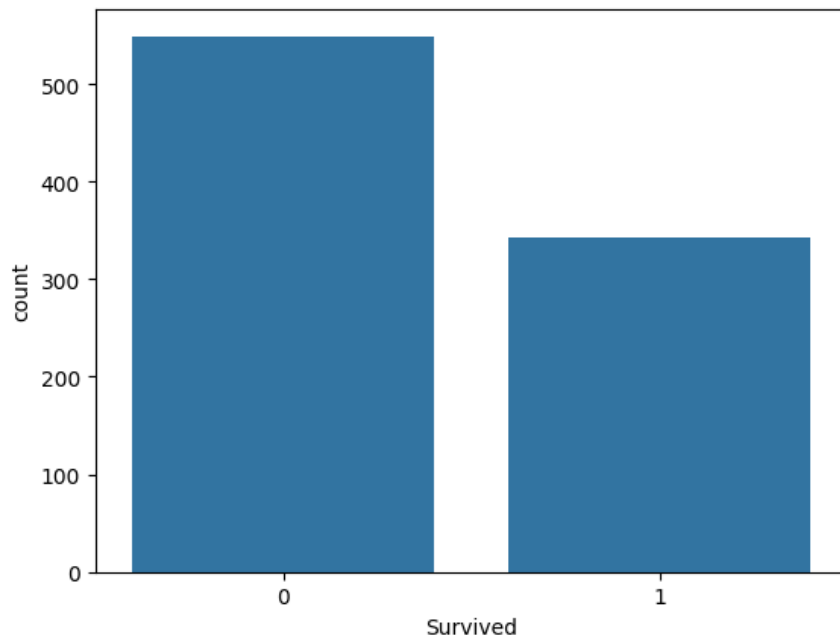


Рис. 1.8. Графік «Кількість пасажирів, що вижили»

Виконаємо візуалізацію кількості пасажирів з посадкою у різних портах (рис. 1.9).

```
# Візуалізація кількості пасажирів з різних портів у навчальному наборі даних  
sns.countplot(x='Embarked', data = train_data)  
plt.show()
```

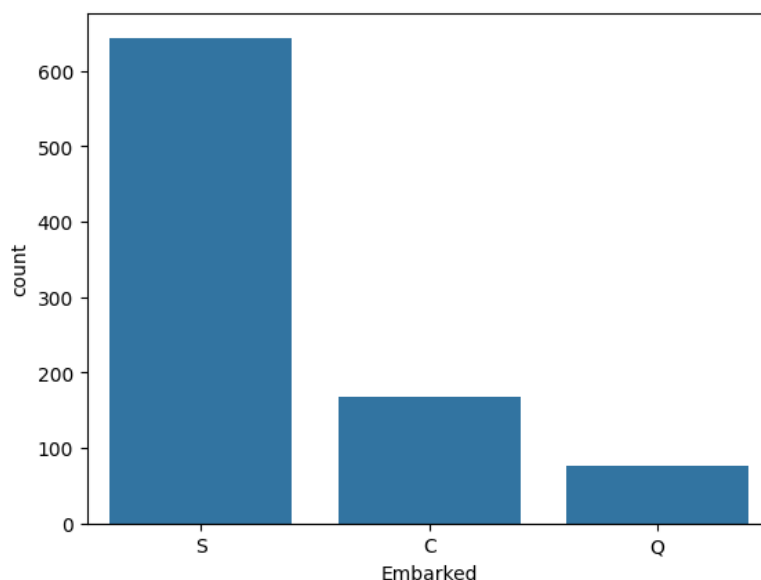


Рис. 1.9. Графік «Кількість пасажирів з різних портів»

Виконаємо візуалізацію впливу гендеру на рівень виживання (рис. 1.10).

```
# Візуалізація впливу гендеру на рівень виживання
sns.countplot(x='Survived', hue = 'Sex', data = train_data)
plt.plot()
```

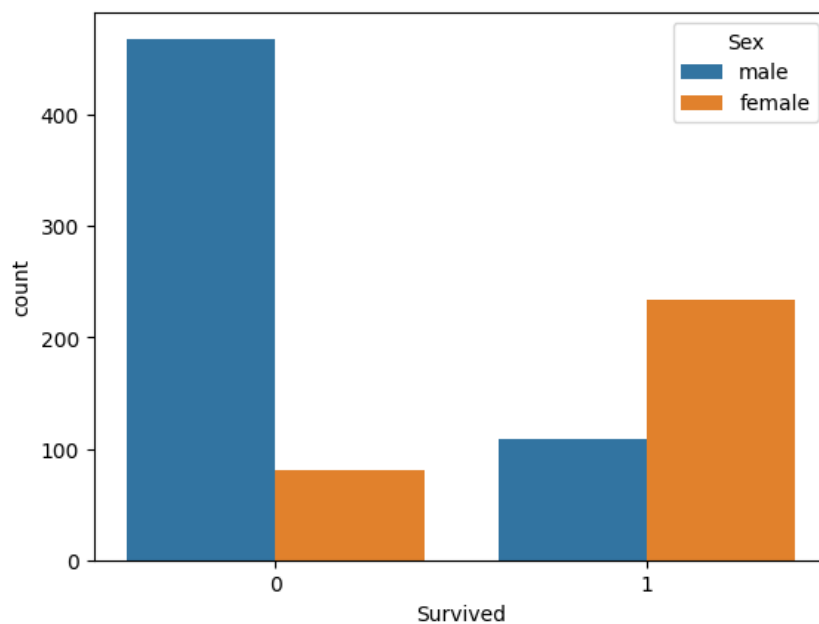


Рис. 1.10. Графік «Вплив гендеру на виживання»

Виконаємо візуалізацію впливу рівня класу (pclass) на рівень виживання (рис. 1.11).

```
# Візуалізація впливу рівня класу (pclass) на рівень виживання
sns.countplot(x="Survived", hue = 'Pclass', data = train_data)
plt.show()
```

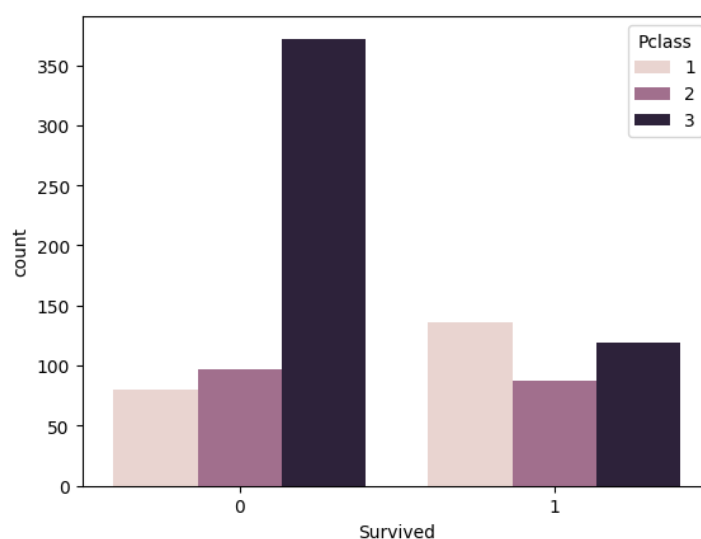


Рис. 1.11. Графік «Вплив рівня класу (pclass) на рівень виживання»

Виконаємо візуалізацію впливу порту посадки на рівень виживання (рис. 1.12).

```
sns.countplot(x='Survived', hue = 'Embarked', data = train_data)
plt.show()
```

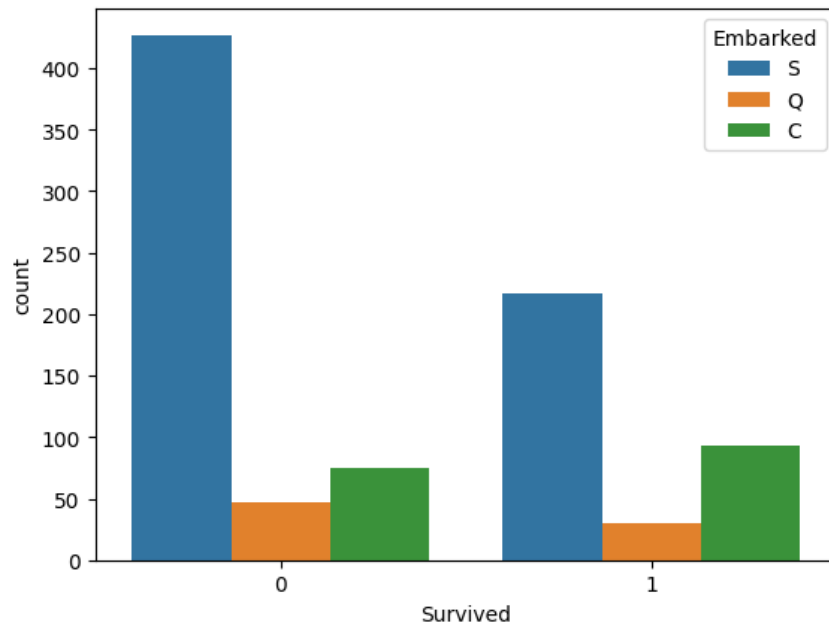


Рис. 1.12. Графік «Вплив порту посадки на рівень виживання»

```
sns.boxplot(x='Fare', data = train_data)
plt.show()
```

Результати свідчать про те, що було дуже мало людей, які заплатили більше 100\$ за квиток

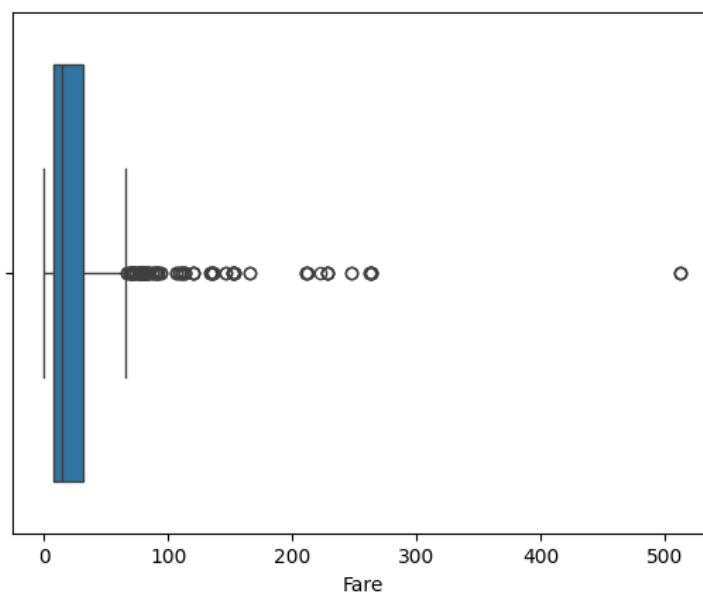


Рис. 1.13. Графік викидів «Fare»

```
sns.boxplot(x='Age', data = train_data)
plt.show()
```

Результати свідчать про те, що в навчальних даних було дуже мало людей, старших за 65 років

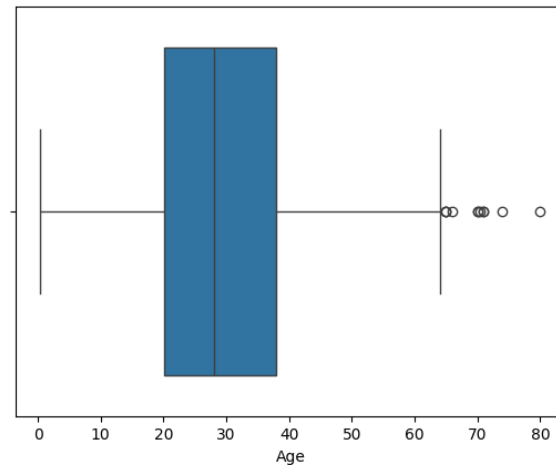


Рис. 1.14. Графік викидів «Age»

```
interval = 10
value_for_bin = np.ceil((train_data.Age.max() - train_data.Age.min()) /
interval).astype(int)
```

```
plt.hist(train_data.Age, bins = value_for_bin)
plt.xlabel("Age")
plt.ylabel("Number")
plt.show()
```

Результати свідчать про те, що більшість пасажирів були у віці від 20 до 40 років

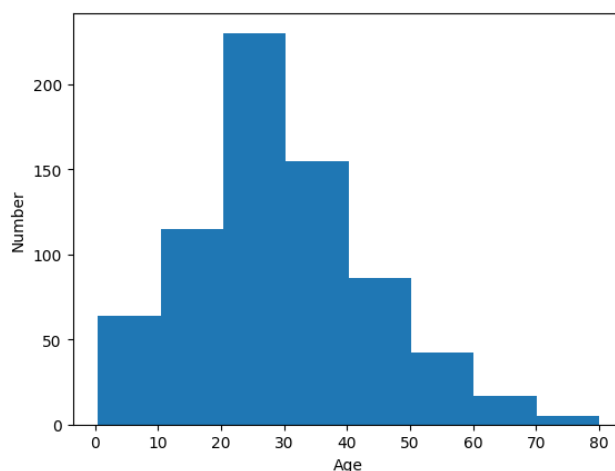


Рис. 1.15. Графік «Віковий діапазон пасажирів»

```
plt.figure(figsize = (10,4))
plt.hist(train_data.Fare, bins = 10, color = 'lime')
plt.xlabel("Fare")
plt.ylabel("Number")
plt.show()
# Результати свідчать, що близько 700 людей заплатили від 0 до 50$
```

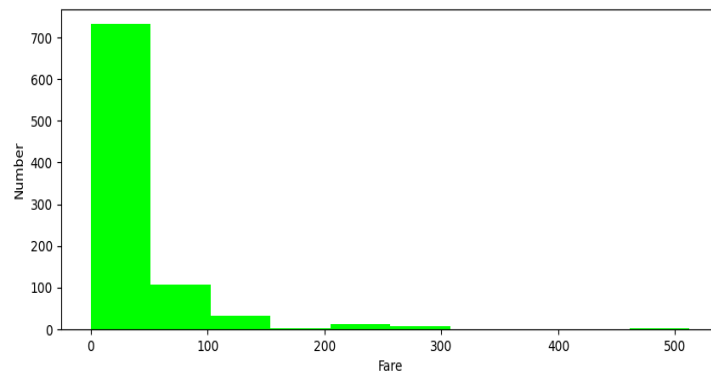


Рис. 1.16. Графік «Витрати на квиток»

```
grid = sns.FacetGrid(train_data, col='Survived', row='Pclass', height=2.2,
aspect=1.6)
grid.map(plt.hist, 'Age', alpha=.5, bins=20)
grid.add_legend()
plt.show()
```

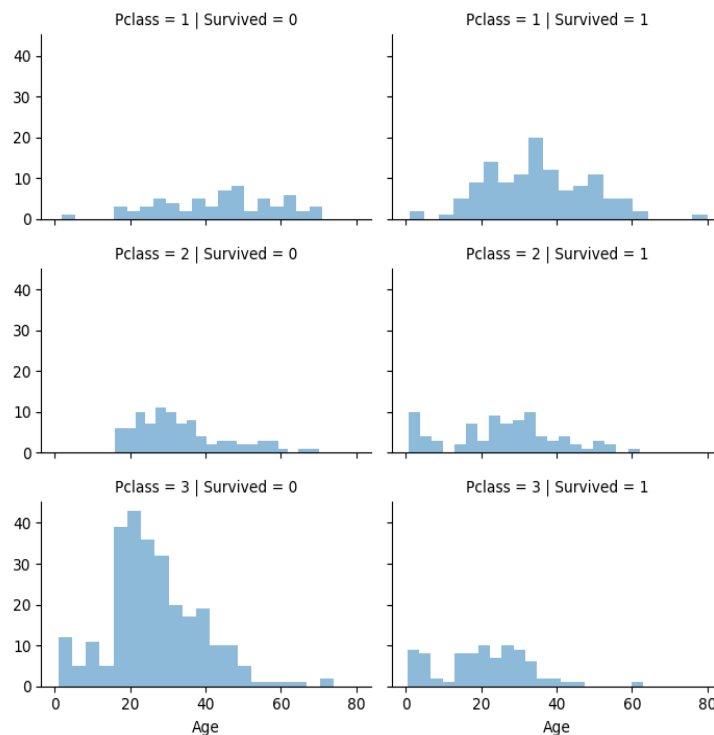


Рис. 1.17. Графіки співвідношення класу та виживання

```

grid = sns.FacetGrid(train_data, row='Embarked', col='Survived', height=2.2,
aspect=1.6)
grid.map(sns.barplot, 'Sex', 'Fare', alpha=.5, ci=None)
grid.add_legend()
plt.show()

```

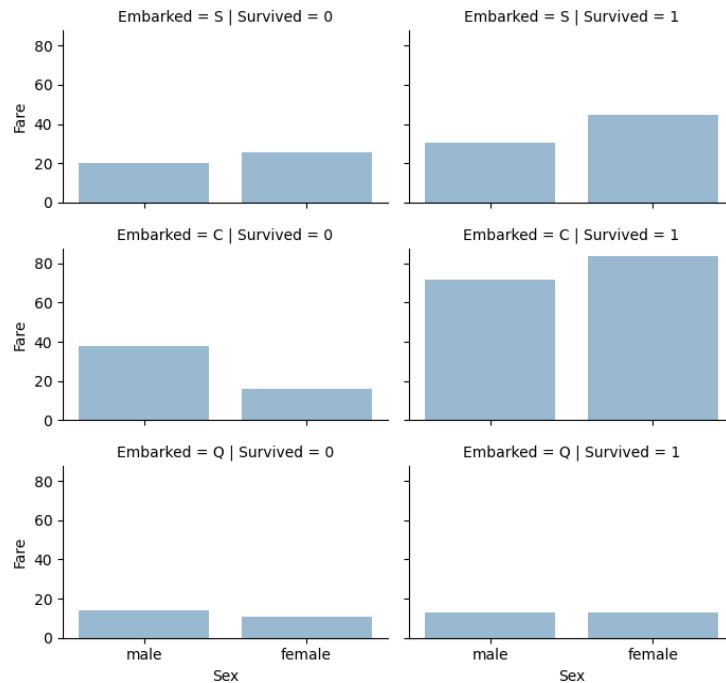


Рис. 1.18. Графіки співвідношення місця посадки та виживання

```

corr_train = train_data.corr()
sns.heatmap(corr_train)
plt.show()

```

Результати свідчать, що стовпці SibSp та Parch видалено, тому ми можемо об'єднати їх для зменшення розмірності набору даних.

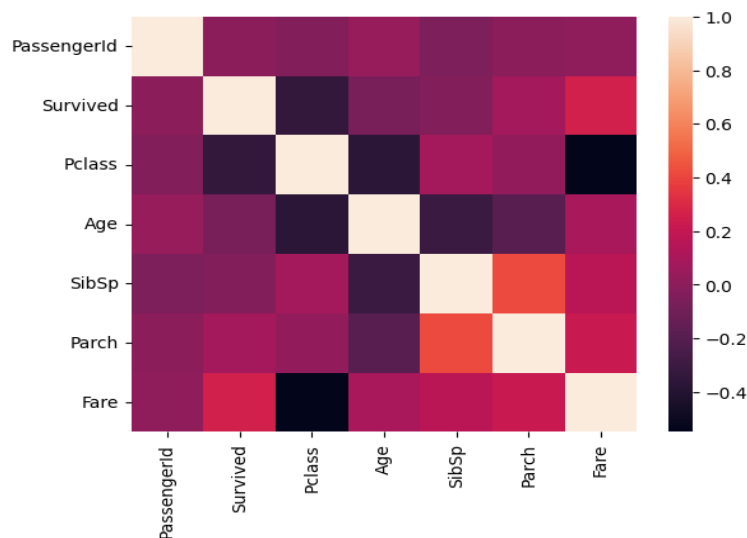


Рис. 1.19. Теплова карта кореляційного зв'язку

Виконаємо простий аналіз даних. Результати свідчать, що жінки мали близько 74% на виживання, в той час як чоловіки мали 81% на смерть (рис. 1.20).

```
((train_data.groupby(['Sex','Survived']).Survived.count() * 100) /
train_data.groupby('Sex').Survived.count())
```

```
Sex      Survived
female  0          25.796178
        1          74.203822
male    0          81.109185
        1          18.890815
Name: Survived, dtype: float64
```

Рис. 1.20. Гендерне розподілення виживання

Результати свідчать, що люди, які належали до третього класу (бідніший), швидше за все, помруть, тоді як люди, які належать до першого класу (багатші), швидше за все, виживуть (рис. 1.21).

```
(train_data.groupby(['Pclass','Survived']).Survived.count() * 100) /
train_data.groupby('Pclass').Survived.count()
```

```
Pclass  Survived
1        0          37.037037
        1          62.962963
2        0          52.717391
        1          47.282609
3        0          75.763747
        1          24.236253
Name: Survived, dtype: float64
```

Рис. 1.21. Класове розподілення виживання

Результати свідчать, про те, що люди, які вирушили з порту Southampton, швидше за все, загинуть (рис. 1.22)

```
(train_data.groupby(['Embarked','Survived']).Survived.count() * 100) /
train_data.groupby('Embarked').Survived.count())
```

```
Embarked  Survived
C          0          44.642857
          1          55.357143
Q          0          61.038961
          1          38.961039
S          0          66.304348
          1          33.695652
Name: Survived, dtype: float64
```

Рис. 1.22. Географічне розподілення виживання

Результати свідчать, що середній вік людей, які вижили, становив близько 28 років (рис. 1.23).

```
train_data.groupby(by=['Survived']).mean()["Age"]
```

Survived	Age
0	30.626179
1	28.343690

Name: Age, dtype: float64

Рис. 1.23. Середній вік осіб, що вижили та померли

Виконаємо роботу над NULL значенням (рис. 1.24). Перед тим, як заповнити відсутні значення, видалимо стовпець Cabin з обох наборів даних.

```
train_data.drop('Cabin', axis = 1, inplace = True)
test_data.drop('Cabin', axis = 1, inplace = True)
combined_data = [train_data, test_data]
for data in combined_data:
    print(data.isnull().sum())
    print('*' * 20)
```

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	2

dtype: int64

PassengerId	0
Pclass	0
Name	0
Sex	0
Age	86
SibSp	0
Parch	0
Ticket	0
Fare	1
Embarked	0

dtype: int64

Рис. 1.24. Визначення кількості NULL значень за стовпцями

Заповнимо стовпчики Age та Fare середніми значеннями, а стовпчик Embarked – значеннями, які зустрічаються найчастіше.

```
for data in combined_data:
    data.Age.fillna(data.Age.mean(), inplace = True)
```

```
data.Fare.fillna(data.Fare.mean(), inplace = True)
# З візуалізацій нам відомо, що порт Southampton є найпопулярнішим
місцем посадки, тому заповнимо пропущені значення "S"

train_data.Embarked.fillna('S', inplace = True)
```

Конвертуємо категоріальні ознаки. Почнемо з перетворення ознаки Sex на категоричні значення female = 1, male = 0.

```
def change_gender(x):
    if x == 'male':
        return 0
    elif x == 'female':
        return 1
train_data.Sex = train_data.Sex.apply(change_gender)
test_data.Sex = test_data.Sex.apply(change_gender)

# або використаємо
# train_data.Sex = train_data.Sex.map({'female':1, 'male':0})
```

За допомогою функції map замінимо значення стовпчика Embarked S = 1, C = 2, Q = 0.

```
change = {'S':1, 'C':2, 'Q':0}
train_data.Embarked = train_data.Embarked.map(change)
test_data.Embarked = test_data.Embarked.map(change)
```

Виконаємо видалення особливостей. Під час візуалізації кореляційної теплової карти ми виявили, що стовпці SibSp та Parch тісно пов'язані між собою, тому створимо новий стовпець під назвою Alone, використовуючи ці два стовпці -----> 1 = Alone , 0 = not Alone.

```
train_data['Alone'] = train_data.SibSp + train_data.Parch
test_data['Alone'] = test_data.SibSp + test_data.Parch

train_data.Alone = train_data.Alone.apply(lambda x: 1 if x == 0 else 0)
test_data.Alone = test_data.Alone.apply(lambda x: 1 if x == 0 else 0)

# Тепер видалимо стовпці SibSp та Parch для навчального та тестового
наборів даних
train_data.drop(['SibSp', 'Parch'], axis = 1, inplace = True)
test_data.drop(['SibSp', 'Parch'], axis = 1, inplace = True)
```

Створимо новий елемент Title назви з існуючого елемента Name (рис. 1.25–1.26).

```
train_data.Name.str.extract('([A-Za-z]+)\.', expand=False).unique().size
# всього 17 унікальних назв

# Створимо елемент Title, що буде містити назву пасажирів, вилучимо
# стовпець Name
for data in combined_data:
    data['Title'] = data.Name.str.extract('([A-Za-z]+)\.', expand = False)
    data.drop('Name', axis = 1, inplace = True)

train_data.Title.value_counts()
```

Mr	517
Miss	182
Mrs	125
Master	40
Dr	7
Rev	6
Mlle	2
Major	2
Col	2
Countess	1
Capt	1
Ms	1
Sir	1
Lady	1
Mme	1
Don	1
Jonkheer	1

Name: Title, dtype: int64

Рис. 1.25. Визначення популярності значень Title

```
test_data.Title.unique()

array(['Mr', 'Mrs', 'Miss', 'Master', 'Ms', 'Col', 'Rev', 'Dr', 'Dona'],
      dtype=object)
```

Рис. 1.26. Масив значень Title

Замінімо назви, що зустрічаються найменше, на Rare.

```
least_occurring = [ 'Don', 'Rev', 'Dr', 'Mme', 'Ms',
                    'Major', 'Lady', 'Sir', 'Mlle', 'Col', 'Capt', 'Countess', 'Dona',
                    'Jonkheer']
for data in combined_data:
    data.Title = data.Title.replace(least_occurring, 'Rare')
# Виконаємо відображення заголовків, замінивши їх на порядкові номери
```

```
title_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}
for data in combined_data:
    data["Title"] = data["Title"].map(title_mapping)
```

Видалимо стовпці PassengerId та Ticket.

```
columns_to_drop = ['PassengerId', 'Ticket']
train_data.drop(columns_to_drop, axis = 1, inplace = True)
test_data.drop(columns_to_drop[1], axis = 1, inplace = True)
```

Об'єднаємо стовпці Age та Fare.

```
for dataset in combined_data:
    dataset.loc[ dataset['Age'] <= 16, 'Age'] = 0
    dataset.loc[(dataset['Age'] > 16) & (dataset['Age'] <= 32), 'Age'] = 1
    dataset.loc[(dataset['Age'] > 32) & (dataset['Age'] <= 48), 'Age'] = 2
    dataset.loc[(dataset['Age'] > 48) & (dataset['Age'] <= 64), 'Age'] = 3
    dataset.loc[ dataset['Age'] > 64, 'Age'] = 4
```

```
for data in combined_data:
    data.loc[data['Fare'] < 30, 'Fare'] = 1
    data.loc[(data['Fare'] >= 30) & (data['Fare'] < 50), 'Fare'] = 2
    data.loc[(data['Fare'] >= 50) & (data['Fare'] < 100), 'Fare'] = 3
    data.loc[(data['Fare'] >= 100), 'Fare'] = 4
```

```
corr_train = train_data.corr()
sns.heatmap(corr_train)
plt.show()
```

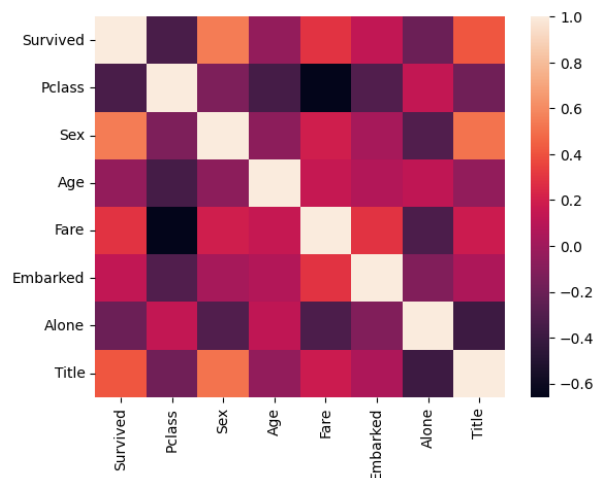


Рис. 1.27. Результуюча теплова мапа

Виконаємо підготовку даних для навчання та тестування.

```
X_train = train_data.drop("Survived", axis=1)
Y_train = train_data["Survived"]
X_test = test_data.drop("PassengerId", axis = 1)
print("shape of X_train",X_train.shape)
print("Shape of Y_train",Y_train.shape)
print("Shape of x_test",X_test.shape)
```

Розробимо та охарактеризуємо модель нейронної мережі (рис. 1.28).

```
# Нейронна мережа
model = Sequential([
    Dense(32, input_dim=7, activation='relu'),
    Dense(64, activation='relu', kernel_initializer='he_normal', use_bias=False),
    tf.keras.layers.BatchNormalization(),
    Dense(128, activation='relu', kernel_initializer='he_normal', use_bias=False),
    Dropout(0.1),
    Dense(64, activation='relu', kernel_initializer='he_normal', use_bias=False),
    Dropout(0.1),
    Dense(32, activation='relu'),
    Dropout(0.15),
    Dense(16, activation='relu'),
    Dense(8, activation='relu', kernel_initializer='he_normal', use_bias=False),
    Dense(1, activation='sigmoid')
])

model.summary()
```

Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	256
dense_1 (Dense)	(None, 64)	2048
batch_normalization (Batch Normalization)	(None, 64)	256
dense_2 (Dense)	(None, 128)	8192
dropout (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8192
dropout_1 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 32)	2080
dropout_2 (Dropout)	(None, 32)	0
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 8)	128
dense_7 (Dense)	(None, 1)	9
=====		
Total params: 21689 (84.72 KB)		
Trainable params: 21561 (84.22 KB)		
Non-trainable params: 128 (512.00 Byte)		
=====		

Рис. 1.28. Характеристика моделі нейронної мережі

Скомпілюємо та натренуємо модель (рис. 1.29).

```
model.compile(loss=tf.keras.losses.binary_crossentropy,
              optimizer=tf.keras.optimizers.Adam(),
              metrics=['accuracy'])
model.fit(X_train, Y_train, batch_size=32, verbose=2, epochs=50)
```

```
Epoch 43/50
28/28 - 0s - loss: 0.3730 - accuracy: 0.8462 - 151ms/epoch - 5ms/step
Epoch 44/50
28/28 - 0s - loss: 0.3737 - accuracy: 0.8361 - 201ms/epoch - 7ms/step
Epoch 45/50
28/28 - 0s - loss: 0.3668 - accuracy: 0.8339 - 172ms/epoch - 6ms/step
Epoch 46/50
28/28 - 0s - loss: 0.3648 - accuracy: 0.8406 - 195ms/epoch - 7ms/step
Epoch 47/50
28/28 - 0s - loss: 0.3678 - accuracy: 0.8429 - 248ms/epoch - 9ms/step
Epoch 48/50
28/28 - 0s - loss: 0.3677 - accuracy: 0.8361 - 129ms/epoch - 5ms/step
Epoch 49/50
```

Рис. 1.29. Процес навчання нейронної мережі

Прогнозування на основі тестових даних (рис. 1.30–1.32).

```
predict = model.predict(X_test)
predict = (predict > 0.5).astype(int).ravel()
print(predict)
```

```
14/14 [=====] - 1s 4ms/step
[0 1 0 0 1 0 1 0 1 0 0 1 1 0 1 1 0 0 0 1 0 1 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0
 0 0 1 0 0 0 1 1 0 1 0 1 1 1 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 1 0 0 1
 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0
 1 1 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0
 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 1
 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0
 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 1
 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 0 0
 1 0 1 0 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0
 1 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0
 0 1 0 0 1 1 0 0 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0
 0 1 1 1 1 0 0 1 0 0 1]
```

Рис. 1.30. Прогнозування «вижив/не вижив»

```
submit = pd.DataFrame({"PassengerId":test_data.PassengerId,
'Survived':predict})
submit.to_csv("final_submission.csv",index = False)

from sklearn import metrics
Y_pred_rand = (model.predict(X_train) > 0.5).astype(int)
print('Precision : ', np.round(metrics.precision_score(Y_train,
Y_pred_rand)*100,2))
print('Accuracy : ', np.round(metrics.accuracy_score(Y_train,
Y_pred_rand)*100,2))
print('Recall : ', np.round(metrics.recall_score(Y_train, Y_pred_rand)*100,2))
print('F1 score : ', np.round(metrics.f1_score(Y_train, Y_pred_rand)*100,2))
print('AUC : ', np.round(metrics.roc_auc_score(Y_train, Y_pred_rand)*100,2))
```

```
28/28 [=====] - 0s 7ms/step
Precision : 80.42
Accuracy : 84.62
Recall : 79.24
F1 score : 79.82
AUC : 83.61
```

Рис. 1.31. Результуючі показники моделі

```
# Побудова Confusion Matrix на тепловій карті
matrix = metrics.confusion_matrix(Y_train, Y_pred_rand)
sns.heatmap(matrix, annot = True,fmt = 'g')
plt.show()
```

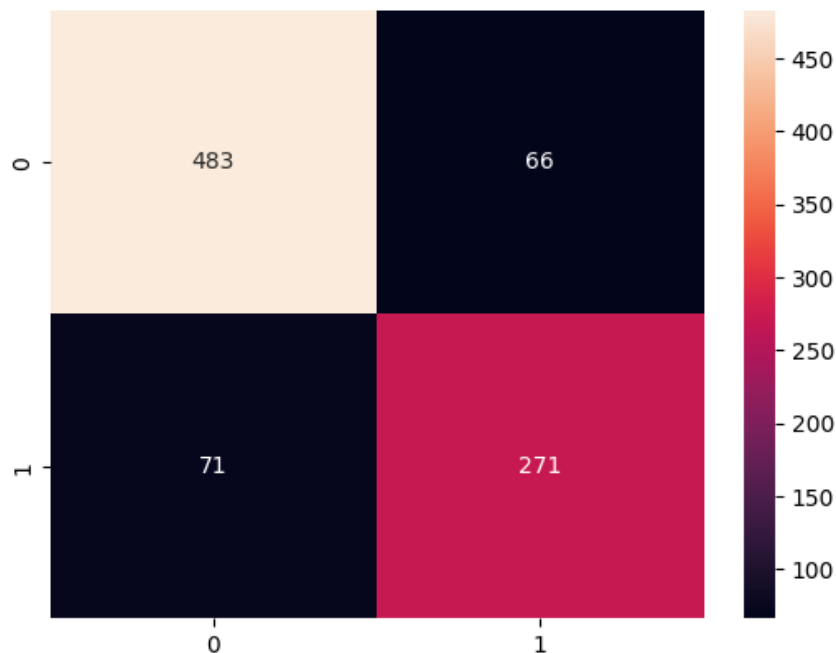


Рис. 1.32. Confusion Matrix моделі

Перелік рекомендованої літератури

1. Булгакова О.С. Методи та системи штучного інтелекту : теорія та практика : навч. посіб. / О.С. Булгакова, В.В. Зосімов, В.О. Поздєєв. – Одеса : ОЛДІ-ПЛЮС, 2020. – 356 с.
2. Гавриленко С.Ю. Машинне навчання : лабораторний практикум / С.Ю. Гавриленко, В.В. Челак, О.А. Горносталь, В.Д Зозуля. – Харків : НТУ «ХПІ», 2022. – 86 с.
3. Кононова К. Ю. Машинне навчання : методи та моделі : підручник. – Харків : ХНУ імені В. Н. Каразіна, 2020. – 301 с.
4. Методи та системи штучного інтелекту : навч. посіб. / Д.В. Лубко, С.В. Шаров. – Мелітополь : ФОП Однорог Т.В., 2019. – 264 с.
5. Субботін С.О. Нейронні мережі : навч. посібник / С.О. Субботін, А.О. Олійник; за ред. С.О. Субботіна. – Запоріжжя : ЗНТУ, 2014. – 132 с.
6. Федорін І.В. Методи та технології обчислювального інтелекту : практикум : навч. посіб. – Київ : КПІ ім. Ігоря Сікорського, 2022. 317 с.
7. Шаповал Н. В. Методи та системи штучного інтелекту. Комп'ютерний практикум : навч. посіб. – Київ : КПІ ім. І. Сікорського, 2022. – 45 с.

8. Charu C. Aggarwal Neural Networks and Deep Learning. Springer International Publishing. 2023. 529 p.
9. Dulani Meedeniya Deep Learning. CRC Press LLC, 2023. 200 p.
10. Foster, D. Generative Deep Learning. O'Reilly Media, Inc. 2023. 456 p.
11. Gallatin, K. & Albon, C. Machine Learning with Python Cookbook. Practical Solutions from Preprocessing to Deep Learning. O'Reilly Media, Inc. 2023. 416 p.
12. Hapke, H., & Nerlson, C. Building Machine Learning Pipelines. Automating Model Life Cycles with TensorFlow. O'Reilly Media, Inc. 2020. 367 p.
13. Howard, J., & Gugger, S. Deep Learning for Coders with fastai & PyTorch. AI Applications Without a PhD. O'Reilly Media, Inc. 2020. 624 p.
14. Huyen, C. Designing Machine Learning Systems. An Iterative Process for Production-Ready Applications. O'Reilly Media, Inc. 2022. 389 p.
15. Huyen, C. Designing Machine Learning Systems. An Iterative Process for Production-Ready Applications. O'Reilly Media, Inc. 2022. 389 p.
16. Lakshmanan, V., Görner, M., & Gillard, R. Practical Machine Learning for Computer Vision. End to-End Machine Learning for images. 2021. 481 p.
17. Machine Learning and Its Application to Reacting Flows : ML and Combustion / N. Swaminathan, A. Parente (eds.). Cham : Springer, 2023. 346 p.
18. Moroney, L. AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence. O'Reilly Media, Inc. 2021. 392 p.

Навчально-методичне видання

МОДЕЛЮВАННЯ ЗАДАЧ ШТУЧНОГО ІНТЕЛЕКТУ

Методичні вказівки
до виконання розрахунково-графічної роботи
для підготовки здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальностями 122 «Комп'ютерні науки»,
126 «Інформаційні системи і технології»

Укладачі: **Гончаренко** Тетяна Андріївна,
Гоц Владислав Володимирович,
Долгополов Сергій Юрійович

Випусковий редактор *Л.С. Тавлуй*
Комп'ютерне верстання *Т.І. Кукарєвої*

Підписано до друку 19.11.2024. Формат 60 × 84_{1/16}
Ум. друк. арк. 1,86. Обл.-вид. арк. 2,0.
Електронний документ. Вид. № 135/III–24.

Видавець і виготовлювач
Київський національний університет будівництва і архітектури

Проспект Повітряних Сил, 31, Київ, Україна, 03037

Свідоцтво про внесення до Державного реєстру суб'єктів
видавничої справи ДК № 808 від 13.02.2002 р.

