

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет будівництва і архітектури

ШТУЧНИЙ ІНТЕЛЕКТ ТА ГІБРИДНІ МЕРЕЖІ

Методичні вказівки до виконання курсової роботи
для здобувачів ОПП другого магістерського рівня вищої освіти
спеціальності 122 «Комп'ютерні науки»

Київ 2024

УДК 004.8
М

Укладачі: С. Ю. Долгополов, асистент

Рецензент І. А. Ачкасов, доктор технічних наук, професор

Відповідальний за випуск Т. А. Гончаренко, канд. техн. наук, доцент,
завідувач кафедри інформаційних технологій

*Затверджено на засіданні кафедри інформаційних технологій,
протокол № 7 від 09 лютого 2024 року.*

В авторській редакції.

М Штучний інтелект та гібридні мережі: методичні вказівки до виконання
курсової роботи / уклад.: С. Ю. Долгополов. – Київ: КНУБА, 2024. – 36 с.

Містять зміст, порядок оформлення і вказівки до виконання курсової роботи.

Призначено для здобувачів ОП другого рівня вищої освіти (магістр)
спеціальності 122 «Комп'ютерні науки».

© КНУБА, 2024

ЗМІСТ

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	4
2. СТРУКТУРА КУРСОВОЇ РОБОТИ.....	5
3. ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	7
4. ЗАВДАННЯ ДЛЯ КУРСОВОЇ РОБОТИ	10
5. ПОРЯДОК ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ.....	13
6. ЗРАЗОК ВИКОНАННЯ ПРАКТИЧНОЇ ЧАСТИНИ ДОСЛІДЖЕННЯ	15
ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	35

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Курсова робота є підсумковим етапом вивчення здобувачами вищої освіти спеціальності 122 «Комп'ютерні науки» дисципліни «Штучний інтелект та гібридні мережі».

При виконанні курсової роботи здобувач вищої освіти повинен продемонструвати вміння самостійно працювати і застосовувати на практиці теоретичні знання, отримані під час вивчення дисципліни.

Метою курсової 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 для підготовки здобувачів до вирішення реальних задач в 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 курсової роботи обумовлена основними розділами дисципліни «Штучний інтелект та гібридні мережі» – 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.

Курсова 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 курсової роботи, що може бути пов'язаною з їхнім науковим дослідженням або професійною діяльністю.

2. СТРУКТУРА КУРСОВОЇ РОБОТИ

Курсова робота складається з наступних розділів:

- вступ;
- аналіз актуальних досліджень та розробок у сфері штучного інтелекту та гібридних мереж;
- 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 аналіз показників якості моделі/моделей;
- висновки: аналіз отриманих результатів, їх значення та можливі напрямки подальших досліджень;
- список використаних джерел;
- додатки: лістинги програм, масивні графічні матеріали тощо (за необхідності).

Обсяг звіту з курсової роботи, незалежно від теми, повинен бути не менше 30-ти сторінок.

Розділи курсового проєкту 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'єкту 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 штучного інтелекту та гібридних мереж 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, їх 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 напрямки для подальших досліджень.

7. Список використаних джерел.

Перелік усієї наукової, методичної літератури та інших джерел, які були використані під час підготовки курсової 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.

3. ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Титульний лист пояснювальної записки оформлюється за зразком, наведеним у додатку. Усі листи записки повинні бути пронумеровані. Першим листом 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. Слід використовувати шрифт Times New Roman 14 розміру з міжрядковим інтервалом 1,5.

1. Рисунок 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, Продовження табл. 1. На наступній сторінці вказують тільки нумерацію граф без елементів головки таблиці, тобто у правому куті пишуть, наприклад, Продовження 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.

4. 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 мовою. До списку використаної літератури рекомендується включати такі джерела: основні нормативні документи і матеріали (державні і 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 курсової роботи потрібно використати не менше 10 джерел. Назви літературних джерел наводять мовою, якою вони написані – українською, англійською та інші. Відомості про літературні джерела подають відповідно до вимог ДСТУ 8302:2015.

5. Презентація

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

4. ЗАВДАННЯ ДЛЯ КУРСОВОЇ РОБОТИ

Задачі для курсової роботи обираються згідно з приведених варіантів:

1. Прогнозування цін на житло за допомогою регресійних моделей.
2. Використання генеративних змагальних нейронних мереж (GANs) для створення синтетичних наборів даних.
3. Аналіз можливостей та обмежень гібридних нейронних мереж у фінансових прогнозах.
4. Аналіз емоцій у текстах з використанням моделей NLP.
5. Розробка гібридної системи для автоматичного розпізнавання та аналізу емоцій на основі тексту та зображень.
6. Розпізнавання об'єктів на зображеннях з використанням CNN.
7. Прогнозування погоди за даними часових рядів.
8. Автоматичне генерування тексту з використанням RNN.
9. Розробка гібридної системи для автоматичного створення музичних композицій з використанням GANs та RNN.
10. Класифікація музичних жанрів за аудіофайлами.
11. Визначення фейкових новин за допомогою машинного навчання.
12. Автоматизований аналіз настроїв у соціальних мережах.
13. Розпізнавання рукописного введення з використанням глибокого навчання.

14. Оптимізація логістичних маршрутів за допомогою алгоритмів машинного навчання.
15. Прогнозування акційних цін за допомогою LSTM мереж.
16. Виявлення шахрайства з кредитними картками.
17. Рекомендаційна система для електронної комерції.
18. Класифікація зображень рентгенівських знімків.
19. Аналіз та прогнозування виборчих трендів.
20. Розпізнавання мови за аудіозаписами.
21. Виявлення дефектів на виробничих лініях за допомогою комп'ютерного зору.
22. Сегментація клієнтської бази за допомогою алгоритмів кластеризації.
23. Класифікації наукових статей.
24. Визначення віку людини за фотографією.
25. Аналіз ринкових трендів за даними соціальних медіа.
26. Оцінка кредитоспроможності за допомогою машинного навчання.
27. Визначення типів забруднення повітря.
28. Прогнозування врожайності сільськогосподарських культур.
29. Моделювання ризиків стихійних лих за даними сенсорів.
30. Виявлення плагіату в текстах.
31. Визначення оптимальних параметрів для виробництва продукції.
32. Прогнозування динаміки розвитку епідемій.
33. Аналіз і оптимізація енергоспоживання будівель.
34. Розробка системи раннього попередження про землетруси.
35. Класифікація видів тканин за допомогою комп'ютерного зору.
36. Виявлення аномалій у фінансових транзакціях.
37. Розробка персоналізованої медичної діагностичної системи.
38. Розпізнавання дорожніх знаків для автономних транспортних засобів.
39. Прогнозування результатів спортивних змагань.
40. Оптимізація складських запасів за допомогою прогнозування попиту.
41. Аналіз звукових сигналів для виявлення дефектів у промисловому

обладнанні.

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

68.Розробка системи розпізнавання та класифікації рослин та тварин.

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

5. ПОРЯДОК ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ

1. Загальні положення.

Оцінювання курсової роботи відбудуватиметься 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 питання, виявляти розуміння основних аспектів дослідження.

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

- **«Відмінно»**, повне правильне виконання роботи у відповідності до зазначених вимог та критеріїв – 28-30 балів;
- **«Дуже добре»**, повне виконання роботи у відповідності 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 проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D		
60-63	E	задовільно	
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

6. ЗРАЗОК ВИКОНАННЯ ПРАКТИЧНОЇ ЧАСТИНИ ДОСЛІДЖЕННЯ

Титанік – британський пасажирський лайнер компанії White Star Line, який затонув у північній частині Атлантичного океану рано 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 членів 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	Cumings, Mrs. John I	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikinen, Miss. Laina	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	S
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	S

Рисунок 1.1. Датасет на Google Drive

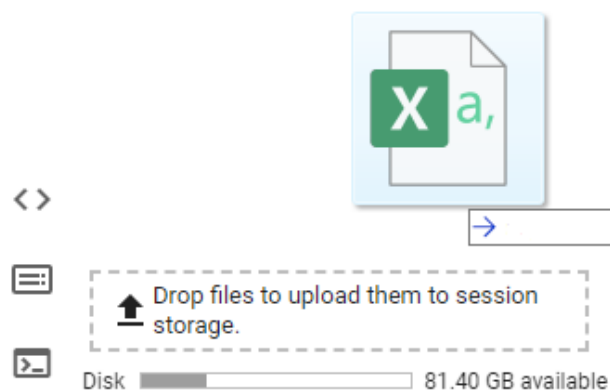


Рисунок 1.2. Завантаження датасету до Google Colab

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
```

Завантаження наборів даних (рис. 1.3–рис. 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

Рисунок 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

Рисунок 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 (Виживші) 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
```

Рисунок 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–рис. 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()
```

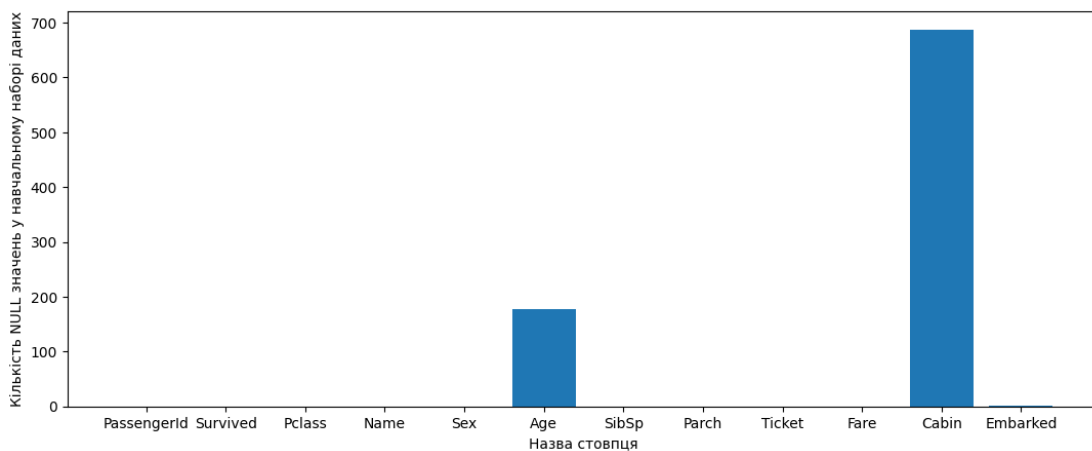


Рисунок 1.6. Графік «Кількість NULL значень у навчальному наборі даних»

```
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()
```

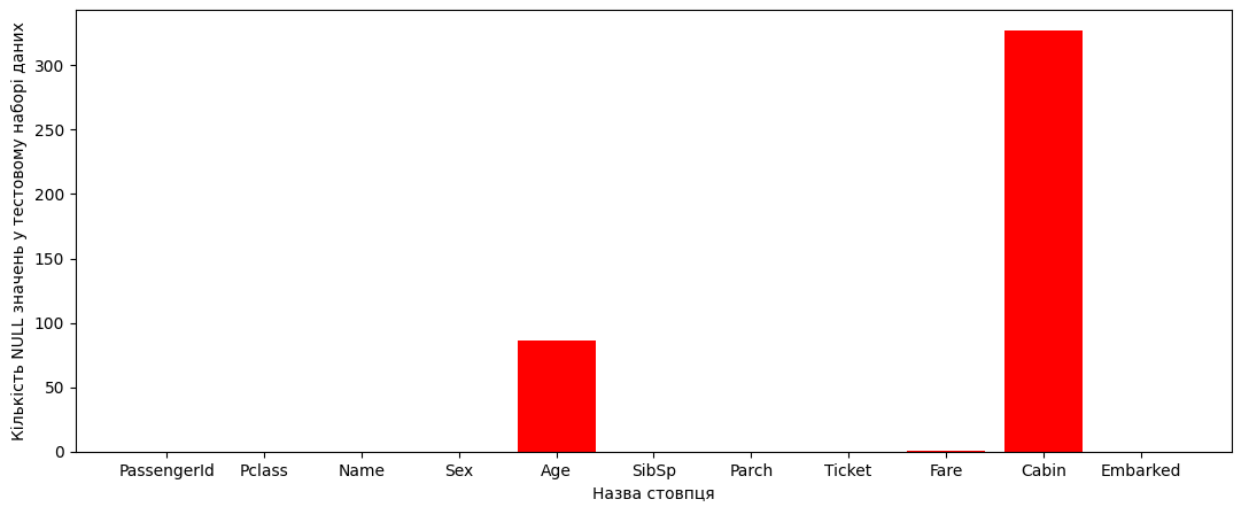


Рисунок 1.7. Графік «Кількість NULL значень у тестовому наборі даних»

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

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

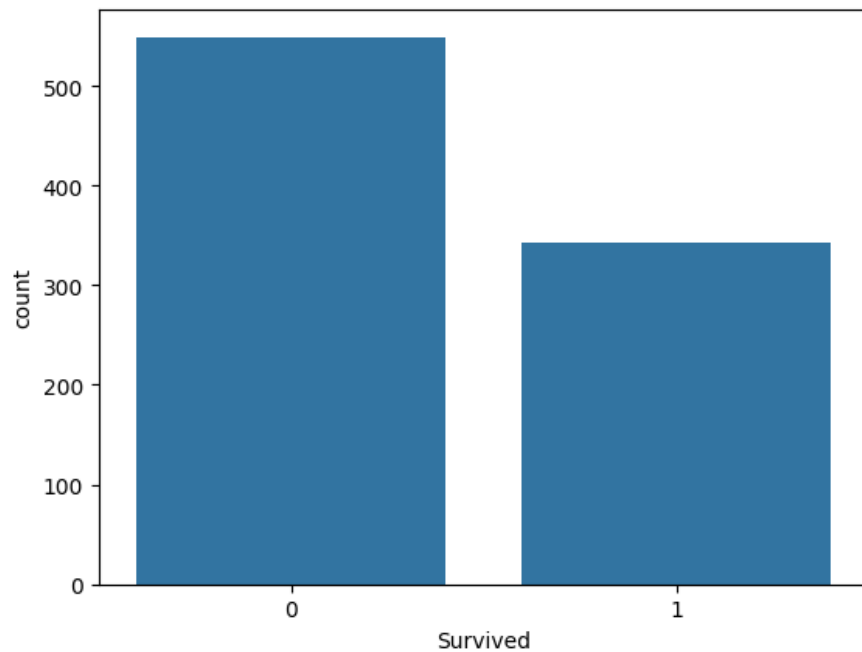


Рисунок 1.8. Графік «Кількість пасажирів, що вижили»

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()
```

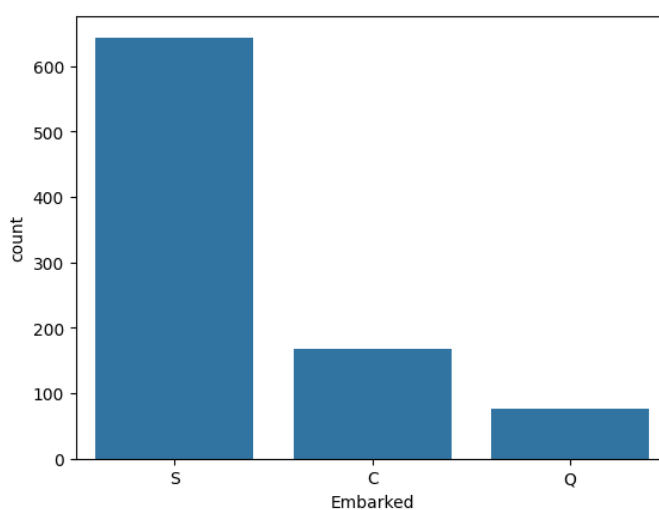


Рисунок 1.9. Графік «Кількість пасажирів з різних портів»

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()
```

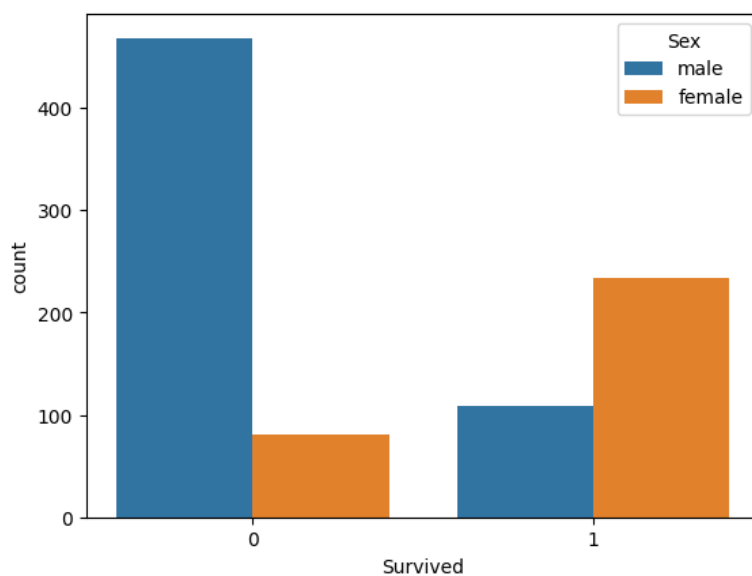


Рисунок 1.10. Графік «Вплив гендеру на виживання»

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()
```

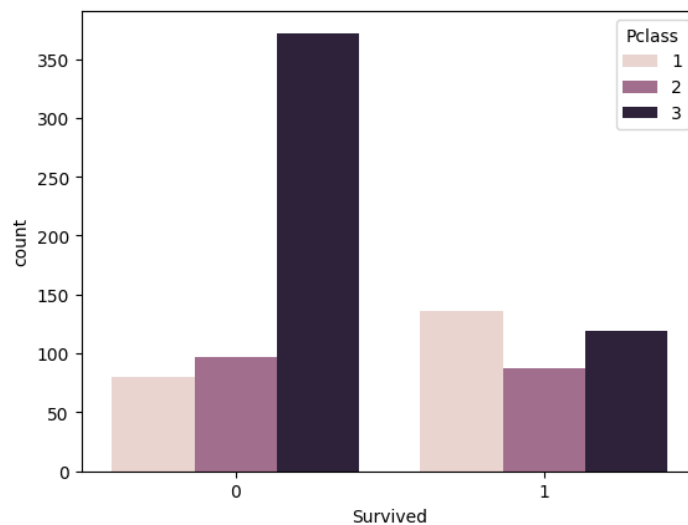


Рисунок 1.11. Графік «Вплив рівня класу (pclass) на рівень виживання»

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

```
# Візуалізація впливу порту посадки на рівень виживання
sns.countplot(x='Survived', hue = 'Embarked', data = train_data)
plt.show()
```

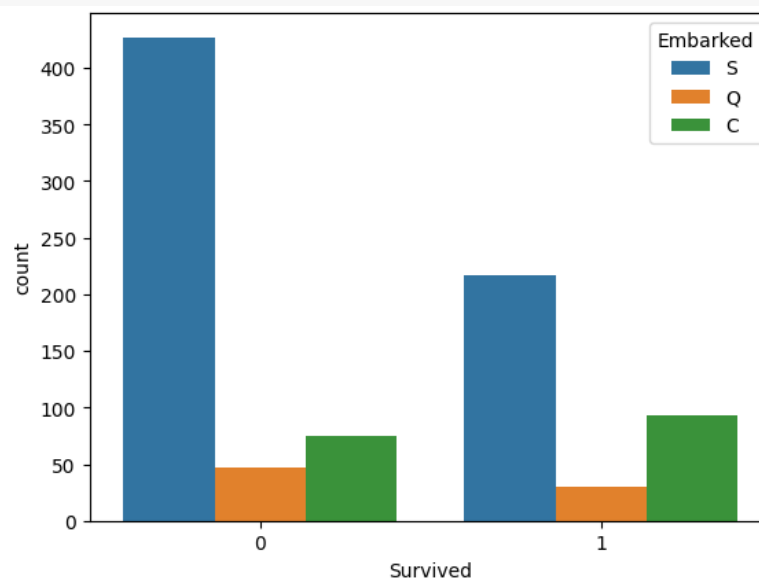


Рисунок 1.12. Графік «Вплив порту посадки на рівень виживання»

```
sns.boxplot(x='Fare', data = train_data)
plt.show()
# Результати свідчать про те, що було дуже мало людей, які заплатили більше 100$ за квиток
```

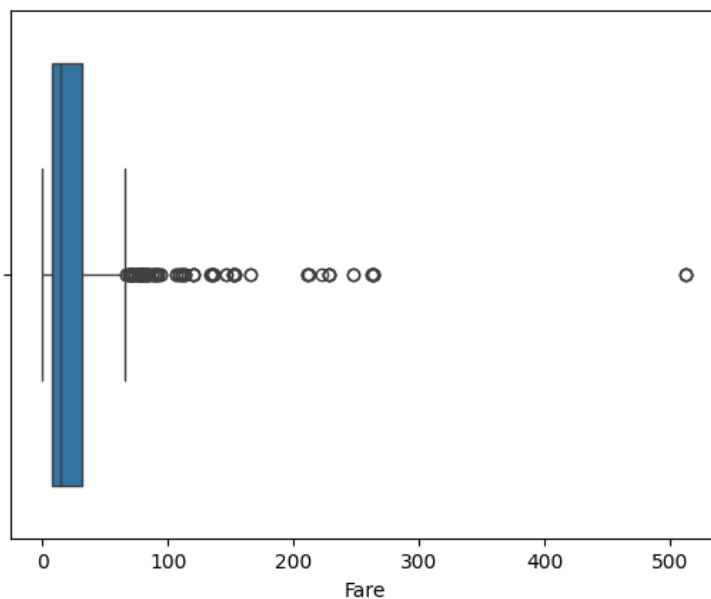


Рисунок 1.13. Графік викидів «Fare»

```
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
```

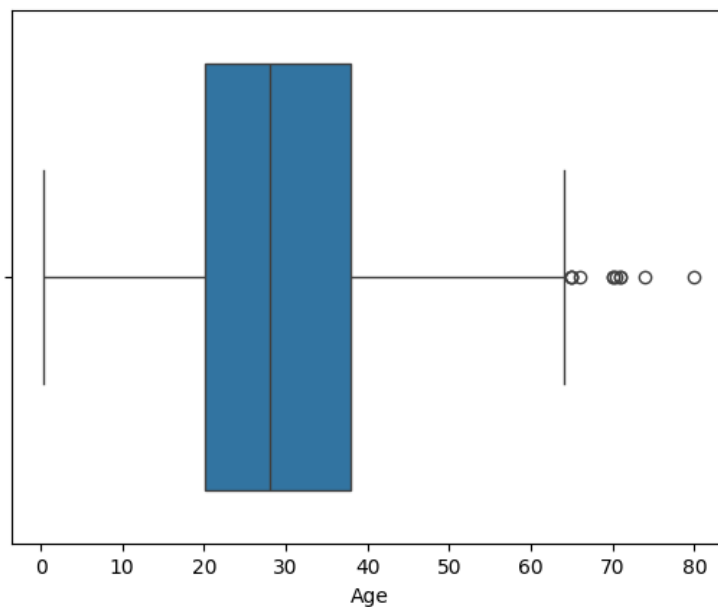


Рисунок 1.14. Графік викидів «Age»

```

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

```

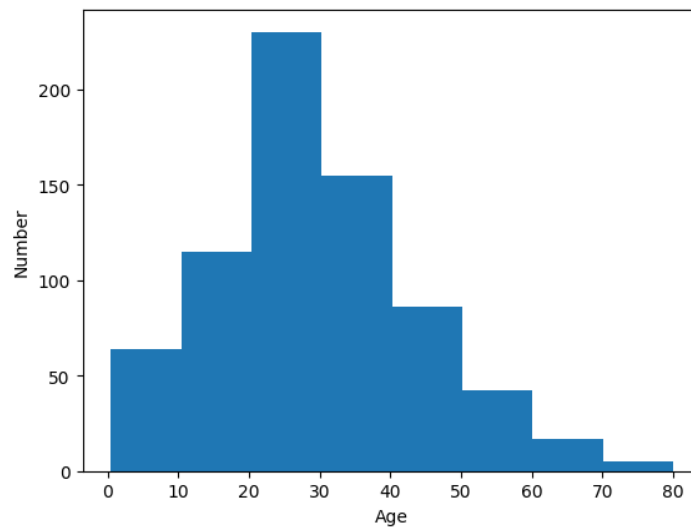


Рисунок 1.15. Графік «Віковий діапазон пасажирів»

```

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$

```

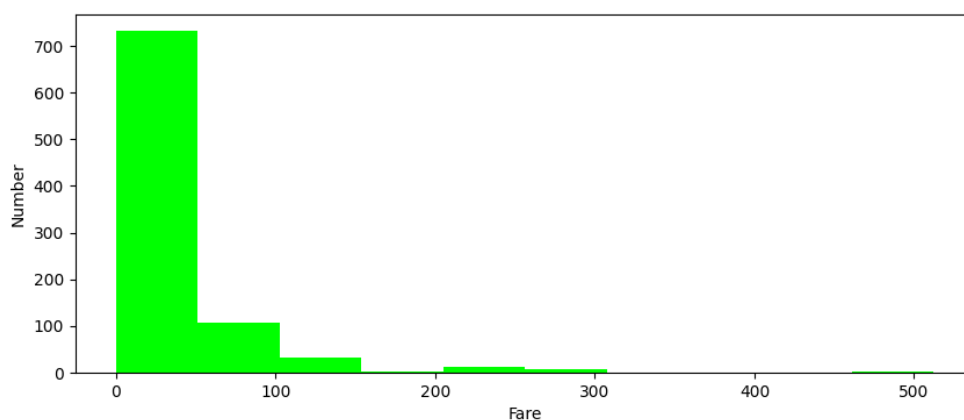


Рисунок 1.16. Графік «Витрати на квиток»

```
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()
```

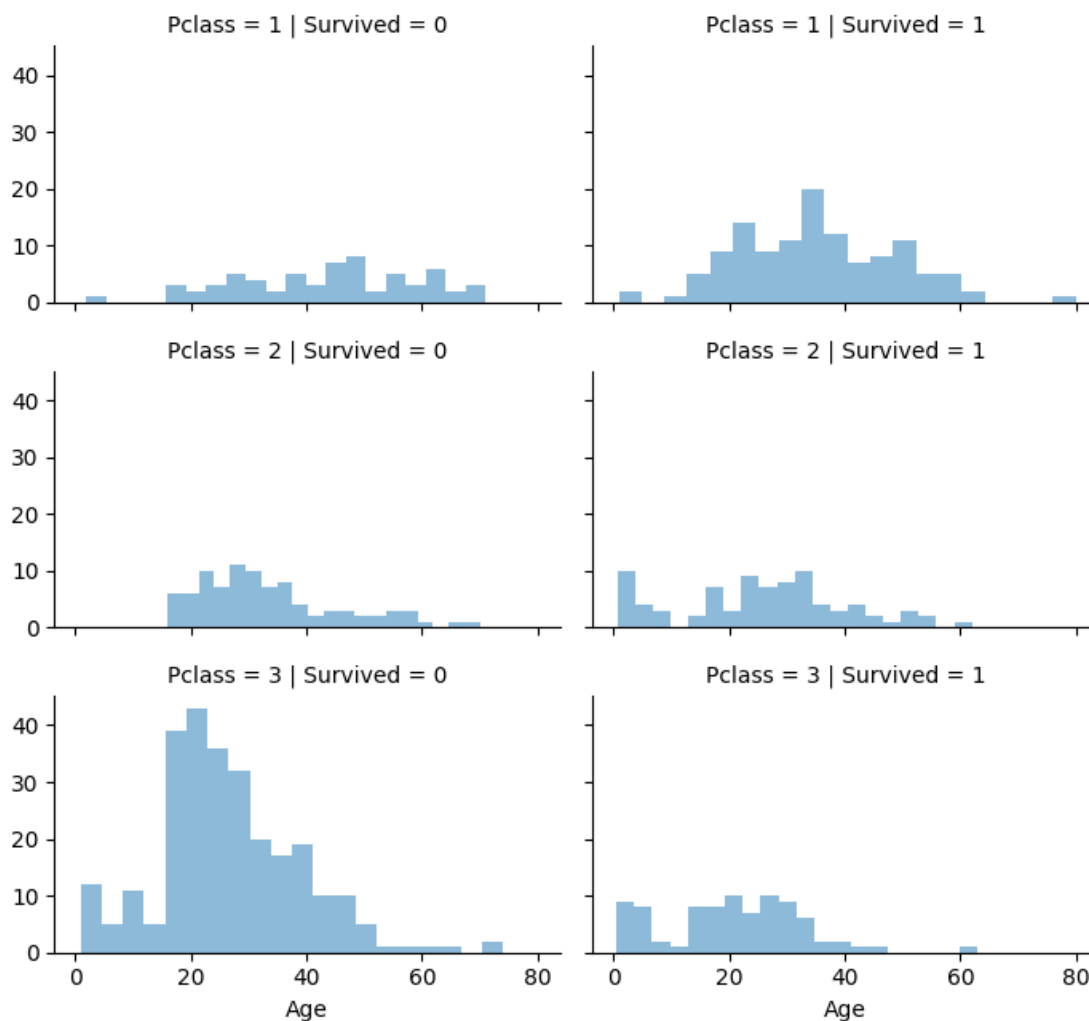


Рисунок 1.17. Графіки співвідношення класу та виживання

```
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()
```

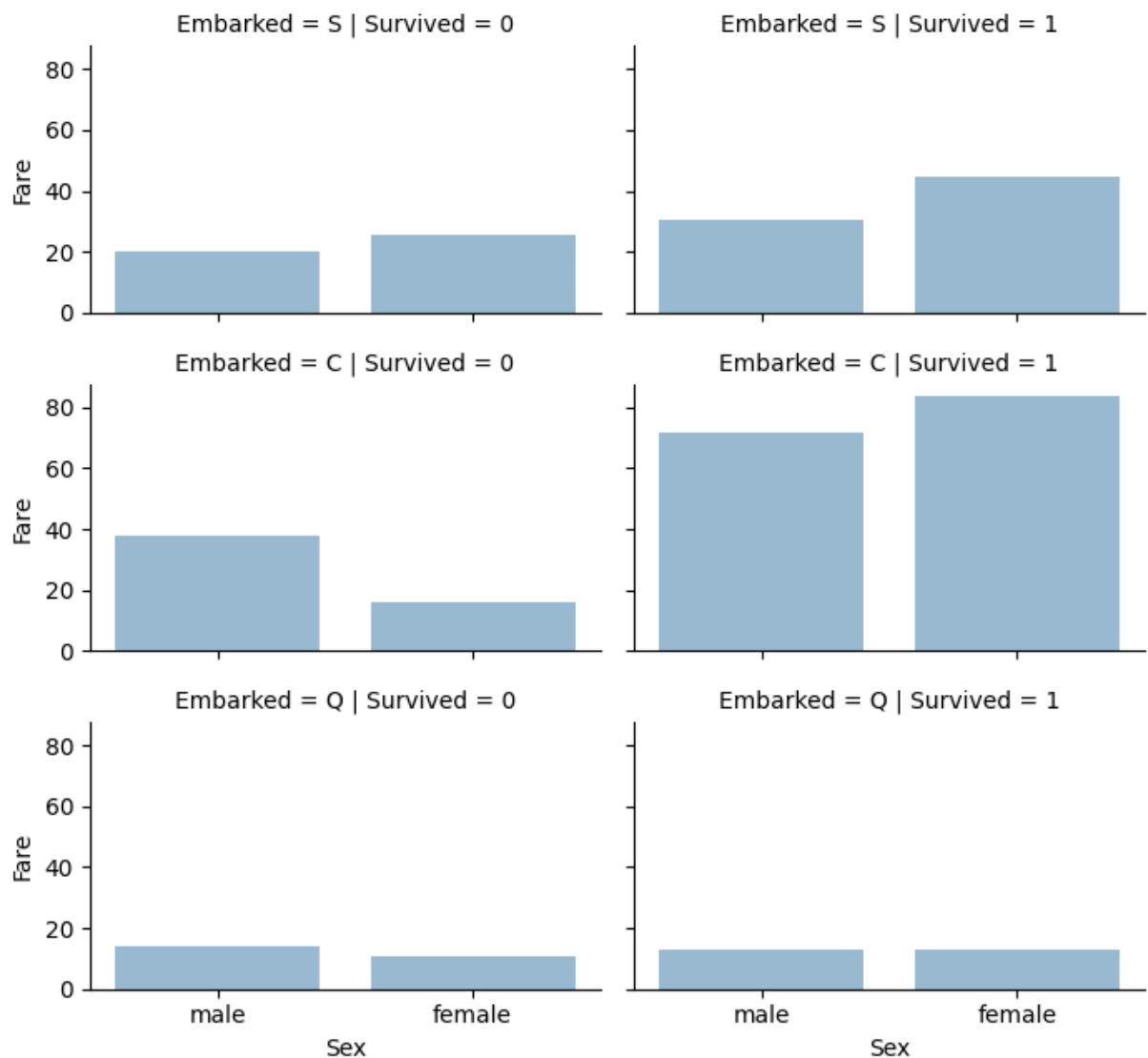


Рисунок 1.18. Графіки співвідношення місця посадки та виживання

```
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.
```

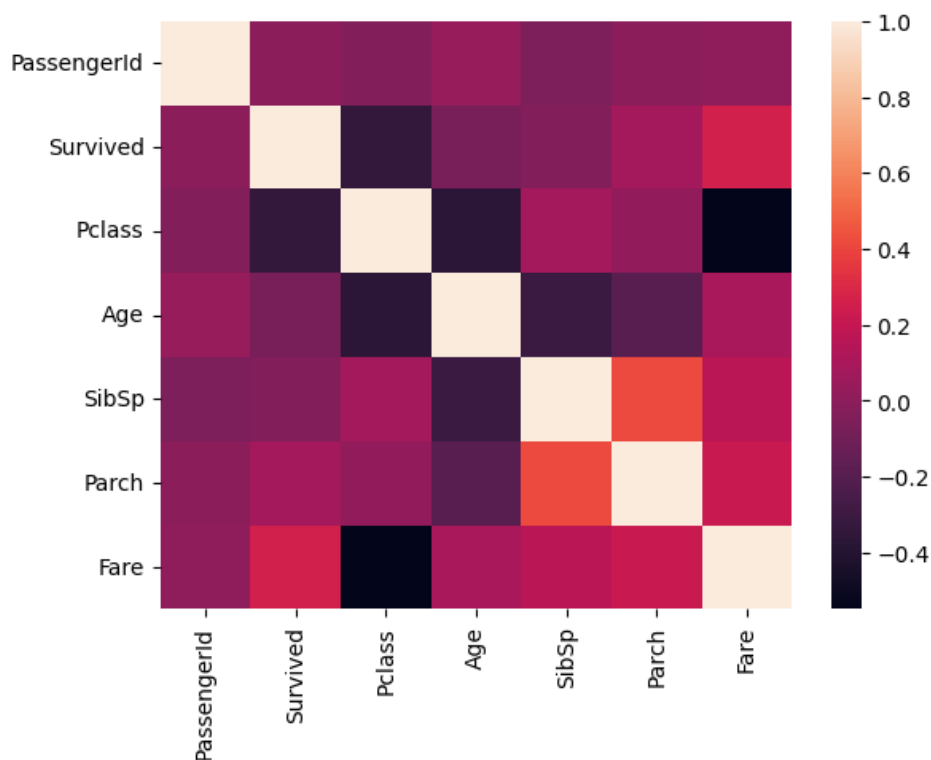


Рисунок 1.19. Теплова карта кореляційного зв'язку

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
```

Рисунок 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
```

Рисунок 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
```

Рисунок 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
0      30.626179
1      28.343690
Name: Age, dtype: float64
```

Рисунок 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
*****
```

Рисунок 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 у 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 стовпчику 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 вилучення 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 вилучимо 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–рис. 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
```

Рисунок 1.25. Визначення популярності значень Title

```
test_data.Title.unique()

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

Рисунок 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()
```

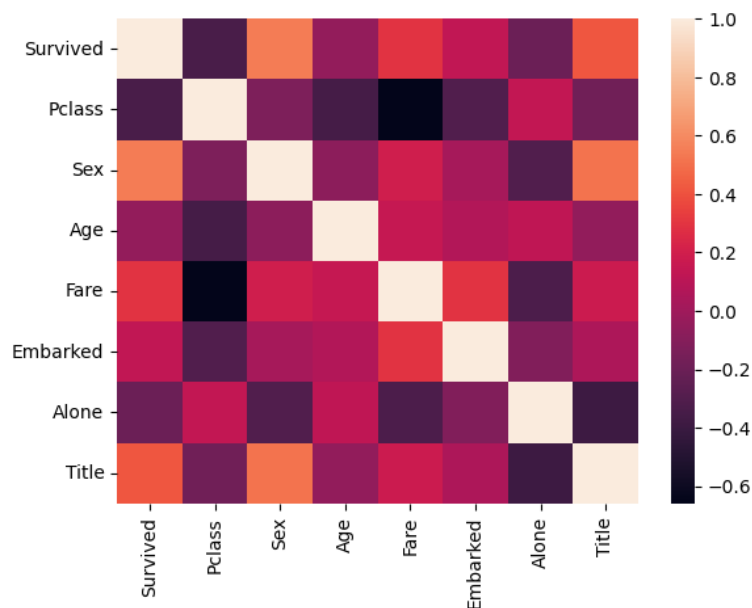


Рисунок 1.27. Результуюча теплова мапа

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)
```

Рисунок 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

Рисунок 1.29. Процес навчання нейронної мережі

Прогнозування на основі тестових даних (рис. 1.30–рис. 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 1 0 0 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 1 0 1 1 1 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 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 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 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]
```

Рисунок 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

```

Рисунок 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()

```

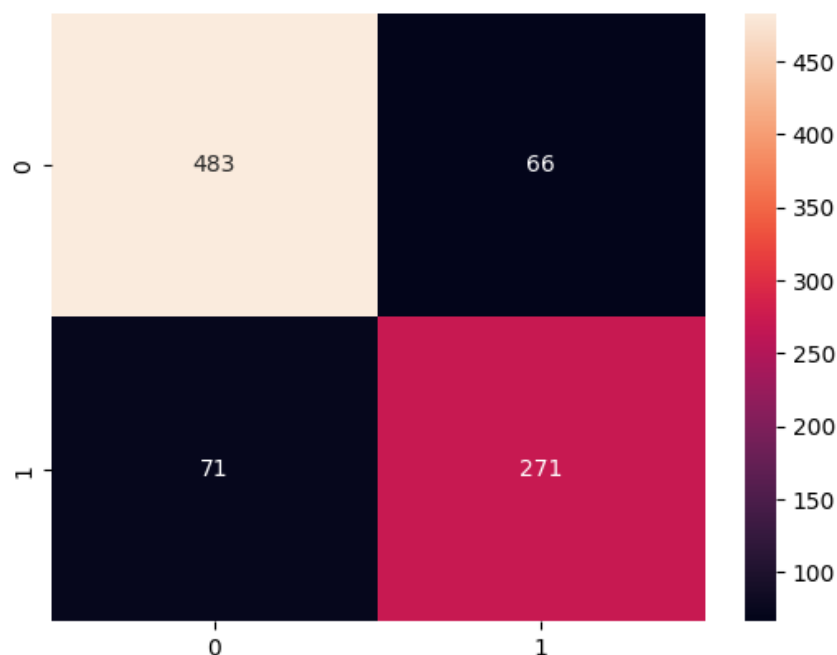


Рисунок 1.32. Confusion Matrix моделі

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Булгакова О.С., Зосімов В.В., Поздєєв В.О. Методи та системи штучного інтелекту:теорія та практика. Навчальний посібник. Одеса : ОЛДІ-ПЛЮС, 2020. 356 с.
2. Гавриленко С.Ю., Челак В.В., Горносталь О.А., Зозуля В.Д. Машинне навчання. Лабораторний практикум / С.Ю. Гавриленко, В.В. Челак, О.А. Горносталь, В.Д Зозуля. – Х.: 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. / укл. 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. Тимошук П.В. Штучні нейронні мережі. Навчальний посібник / Львів: Видавництво Львівської Політехніки, 2011. 444 с.
7. Федорін І. В. Методи та технології обчислювального інтелекту: Практикум : навч. посіб. КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2022. 317 с.
8. Шаповал Н. В. Методи та системи штучного інтелекту. Комп'ютерний практикум : навч. посіб. Київ : КПІ ім. І. Сікорського, 2022. 45 с.
9. Charu C. Aggarwal Neural Networks and Deep Learning. Springer International Publishing. 2023. 529 p.
- 10.Dulani Meedeniya Deep Learning. CRC Press LLC, 2023. 200 p.
- 11.Foster, D. Generative Deep Learning. O'Reilly Media, Inc. 2023. 456 p.
- 12.Gallatin, K. & Albon, C. Machine Learning with Python Cookbook. Practical Solutions from Preprocessing to Deep Learning. O'Reilly Media, Inc. 2023. 416 p.
- 13.Hapke, H., & Nerlson, C. Building Machine Learning Pipelines. Automating Model Life Cycles with TensorFlow. O'Reilly Media, Inc. 2020. 367 p.

14. Howard, J., & Gugger, S. Deep Learning for Coders with fastai & PyTorch. AI Applications Without a PhD. O'Reilly Media, Inc. 2020. 624 p.
15. Huyen, C. Designing Machine Learning Systems. An Iterative Process for Production-Ready Applications. O'Reilly Media, Inc. 2022. 389 p.
16. Huyen, C. Designing Machine Learning Systems. An Iterative Process for Production-Ready Applications. O'Reilly Media, Inc. 2022. 389 p.
17. Lakshmanan, V., Görner, M., & Gillard, R. Practical Machine Learning for Computer Vision. End to-End Machine Learning for images. 2021. 481 p.
18. Machine Learning and Its Application to Reacting Flows : ML and Combustion / N. Swaminathan, A. Parente (eds.). Cham : Springer, 2023. 346 p.
19. Moroney, L. AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence. O'Reilly Media, Inc. 2021. 392 p.
20. Zgurovsky M. Z., Zaychenko Y. P. The Fundamentals of Computational Intelligence: System Approach. Springer International Publishing Switzerland, 2016. 375 p.