

<https://doi.org/10.31891/2219-9365-2025-82-57>

УДК 004.4'2:519.6

ФЛЕГАНТОВ Леонід

Полтавський державний аграрний університет

<https://orcid.org/0000-0002-4689-1457>

e-mail: leonid.flegantov@pdau.edu.ua

ФЛЕГАНТОВА Анна

Полтавський університет економіки і торгівлі

<https://orcid.org/0000-0001-9158-8847>

e-mail: anna.flegantova@gmail.com

ПОНОЧОВНА Олена

Полтавський державний аграрний університет

<https://orcid.org/0000-0002-4377-0633>

e-mail: olena.ponochovna@pdau.edu.ua

ДУГАР Тетяна

Полтавський державний аграрний університет

<https://orcid.org/0000-0002-9642-2763>

e-mail: tetyana.dugar@pdau.edu.ua

СТРУКТУРОВАНІЙ ПІДХІД ДО ВІЗУАЛЬНОГО АНАЛІЗУ ДАНИХ У ЗАДАЧАХ ОБРОБКИ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ PYTHON

Розглянуто структурований підхід до візуального аналізу даних у задачах обробки інформації з використанням бібліотек Python. Проаналізовано особливості бібліотек Matplotlib, seaborn, Pandas Visualization і Plotly, їх сильні та слабкі сторони залежно від типу даних і цілей аналізу. Розроблено методику, що включає п'ять етапів: швидкий первинний огляд, поглиблений одновимірний аналіз, двовимірний і багатовимірний аналіз, цільова візуалізація та підсумкове представлення результатів. Проведено апробацію методики на наборі даних «Титанік», що дозволило виявити ключові фактори, які впливають на виживання пасажирів. Сформовано рекомендації щодо вибору бібліотек для різних типів даних і аудиторій. Запропонований підхід забезпечує ефективність, відтворюваність і економічну доцільність візуалізацій.

Ключові слова: візуальний аналіз, Python, Matplotlib, seaborn, Plotly, обробка даних, візуалізація.

FLEHANTOV Leonid, PONOCHOVNA Olena, DUGAR Tetyana

Poltava State Agrarian University

FLEHANTOVA Anna

Poltava University of Economics and Trade

STRUCTURED APPROACH TO VISUAL DATA ANALYSIS IN INFORMATION PROCESSING TASKS USING PYTHON

This study introduces a structured approach to visual data analysis in information processing tasks using Python libraries, addressing the growing need for effective and reproducible visualization methods in data science. The research systematically evaluates key Python libraries – Matplotlib, seaborn, Pandas Visualization, and Plotly—analyzing their strengths and limitations in relation to data types and analytical objectives. A comprehensive methodology comprising five iterative stages is proposed: rapid preliminary overview, in-depth univariate analysis, bivariate and multivariate analysis, targeted visualization, and final result presentation. These stages ensure systematic data exploration, from identifying initial patterns to generating polished, audience-specific visualizations. The methodology was validated using the "Titanic" dataset, which includes numerical (age, ticket price) and categorical (gender, cabin class) variables. Through the application of the proposed stages, key factors influencing passenger survival were identified, such as gender, cabin class, and ticket price. Visual tools like histograms, scatterplots, heatmaps, and interactive dashboards were employed to uncover trends, correlations, and anomalies, demonstrating the methodology's ability to support data-driven decision-making. Recommendations for selecting appropriate libraries based on data type, analysis goals, and audience were formulated, enhancing the approach's practical utility.

The structured approach integrates theoretical principles of data visualization with practical Python-based tools, offering flexibility, reproducibility, and cost-effectiveness compared to commercial platforms like Tableau. By leveraging open-source libraries, the methodology overcomes limitations in automation and customization, making it suitable for diverse applications in scientific research, business analytics, and technological processes.

Future research directions include integrating the methodology with machine learning techniques to automate graph selection, optimizing library performance for large datasets, and developing standardized templates for automated dashboards. This approach has the potential to advance data visualization practices, supporting digital transformation and efficient information processing across industries.

Keywords: visual data analysis, Python, Matplotlib, seaborn, Plotly, data processing, visualization.

Стаття надійшла до редакції / Received 01.04.2025

Прийнята до друку / Accepted 29.04.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

У сучасних умовах стрімкого зростання обсягів цифрових даних у різних сферах діяльності виникає потреба в ефективних методах їх аналізу та представлення. Візуальний аналіз даних є ключовим інструментом для виявлення закономірностей, тенденцій та аномалій, що сприяє прийняттю обґрунтованих рішень у наукових дослідженнях, бізнесі та технологічних процесах. Проте використання комерційних платформ, таких як Tableau чи Power BI, обмежене високою вартістю, низькою гнучкістю для нестандартних задач та складністю автоматизації. Мова програмування Python, завдяки своїм бібліотекам (Matplotlib, seaborn, Plotly), пропонує гнучке та економічно вигідне рішення для створення інтерактивних і масштабованих візуалізацій, однак потребує систематизованого підходу до вибору інструментів і методів залежно від типу даних і аналітичних задач.

Проблема полягає у відсутності уніфікованої методики, яка б інтегрувала теоретичні принципи візуалізації з практичним застосуванням бібліотек Python, забезпечуючи ефективність і відтворюваність результатів. Розробка такої методики є важливим науковим і практичним завданням, оскільки вона сприяє автоматизації аналітичних процесів, підвищенню якості управлінських рішень і зниженню витрат на програмне забезпечення. Це особливо актуально для інформаційних систем і комп'ютерно-інтегрованих технологій, де обробка великих обсягів даних вимагає швидких, точних і наочних інструментів. Запропонований структурований підхід до візуального аналізу даних засобами Python відповідає сучасним викликам науки про дані, сприяючи цифровій трансформації та оптимізації технологічних процесів.

Метою даного дослідження є розробка структурованого підходу до візуального аналізу даних у задачах обробки інформації з використанням бібліотек Python, що забезпечить ефективність, відтворюваність і економічну доцільність аналітичних процесів. Сучасні виклики науки про дані та цифрової трансформації вимагають гнучких інструментів для швидкого виявлення закономірностей і підтримки управлінських рішень. Запропонований підхід спрямований на подолання обмежень комерційних платформ, таких як Tableau, шляхом використання відкритих бібліотек Python (Matplotlib, seaborn, Plotly) для створення інтерактивних і масштабованих візуалізацій.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Проблематика візуального аналізу даних засобами Python активно досліджується в сучасній науковій літературі, що відображає зростаючу потребу в ефективних інструментах обробки інформації. Бібліотеки Python, такі як Matplotlib [1], seaborn [2], Plotly [3], забезпечують гнучкість і відтворюваність візуалізацій, але їхнє ефективне використання потребує систематизованого підходу. Недостатня кількість уніфікованих методик для вибору інструментів залежно від типу даних і цілей аналізу залишається ключовою проблемою, що ускладнює автоматизацію та інтеграцію в технологічні процеси.

Стаття [4] підкреслює важливість декларативних бібліотек, таких як Altair, для спрощення створення складних візуалізацій, але вказує на їхні обмежену продуктивність із великими даними. Автори дослідження [5] аналізують інтерактивні інструменти (Plotly, Vokeh), наголошуючи на їхній перевазі для веб-додатків, однак відзначають складність навчання для новачків. У дослідженні [6] розглядається інтеграція Python із платформами BI, що частково вирішує проблему автоматизації, але не пропонує економічно доступних рішень. Автори у [7] акцентують на важливості структуризації етапів візуального аналізу для підвищення ефективності, зокрема для геопросторових даних із GeoPandas. Дослідники в [8] пропонують методику комбінування Matplotlib і Plotly для наукових візуалізацій, але не охоплюють категоріальні дані.

Інтернет-джерела, такі як medium.com [9], підкреслюють практичну цінність seaborn для статистичного аналізу, але відзначають потребу в додаткових налаштуваннях через Matplotlib. Stack Overflow [10] містить дискусії про продуктивність Plotly із великими наборами даних. Стаття на kdnuggets.com [11] акцентує на зростанні популярності HoloViz для інтерактивних дашбордів, але бракує порівняння з комерційними аналогами.

Отже, аналіз літератури показує, що хоча окремі аспекти візуалізації в Python добре досліджені, систематизована методика, яка б інтегрувала теоретичні принципи, практичне застосування та економічну ефективність, залишається недостатньо розробленою. Це підтверджує актуальність створення структурованого підходу, орієнтованого на задачі обробки інформації.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Візуальний аналіз даних є ключовим інструментом для обробки інформації в умовах зростання обсягів цифрових даних. Розробка структурованого підходу до візуалізації за допомогою Python дозволяє систематизувати процес аналізу, забезпечити відтворюваність результатів і підвищити економічну ефективність. Візуальний аналіз даних поєднує графічне представлення інформації, інтерактивність і аналітичне мислення для виявлення закономірностей, трендів і аномалій [7]. Основні етапи процесу включають підготовку даних, вибір візуального кодування, генерацію графіків, інтерактивне дослідження та

формулювання висновків. Ці етапи є ітеративними, що дозволяє повертатися до попередніх кроків для уточнення результатів.

Ключові принципи ефективної візуалізації включають точність, відповідність типу графіка даним, максимізацію співвідношення «дані-чорнило», ясність, ефективне використання візуальних змінних (позиція, колір, форма), забезпечення контексту, фокусування на ключових аспектах і візуальну чесність [8]. Наприклад, стовпчикові діаграми підходять для порівняння категорій, а лінійні графіки – для відображення часових трендів.

Python пропонує широкий спектр бібліотек для візуалізації, серед яких Matplotlib, seaborn, Pandas Visualization і Plotly є найбільш універсальними [10]. У табл. 1 наведено порівняльну характеристику цих бібліотек.

Таблиця 1.

Порівняльна характеристика бібліотек Python для візуалізації

Бібліотека	Тип	Сильні сторони	Недоліки	Основне застосування
Matplotlib	Низькорівнева	Гнучкість, контроль над деталями	Складний синтаксис, застарілий дизайн	Наукові публікації, статичні графіки
Pandas Visualization	Високорівнева	Швидкість, інтеграція з Pandas	Обмежена гнучкість	Розвідувальний аналіз
seaborn	Високорівнева	Статистичні графіки, естетика	Обмежена інтерактивність	Статистичний аналіз
Plotly	Високорівнева	Інтерактивність, веб-інтеграція	Складність для великих даних	Дашборди, інтерактивні звіти

Matplotlib забезпечує максимальний контроль, але вимагає багатослівного коду. Pandas Visualization зручна для швидкого аналізу, але обмежена в налаштуваннях. Seaborn спрощує створення статистичних графіків, тоді як Plotly забезпечує створення інтерактивних веб-візуалізацій [11].

Розроблена методика включає п'ять етапів, які систематизують процес візуалізації даних: швидкий первинний огляд, поглиблений одновимірний аналіз, двовимірний і багатовимірний аналіз, цільова візуалізація та підсумкове представлення результатів. Кожен етап використовує відповідні бібліотеки Python залежно від мети та типу даних.

1. Швидкий первинний огляд передбачає завантаження даних у Pandas DataFrame і створення базових графіків (гістограм, лінійних графіків) за допомогою вбудованого функціоналу Pandas Visualization для оцінки їх структури та розподілів. Наприклад, гістограми числових змінних дозволяють виявити зміщення чи викиди.

2. Поглиблений одновимірний аналіз фокусується на детальному вивченні розподілів за допомогою seaborn (histplot, boxplot, countplot). Цей етап допомагає ідентифікувати центральну тенденцію, розкид і аномалії. Matplotlib використовується для додаткового налаштування візуалізації.

3. Двовимірний і багатовимірний аналіз досліджує взаємозв'язки між змінними. Тут основним інструментом є бібліотека seaborn (scatterplot, heatmap, pairplot), тоді як Plotly забезпечує інтерактивність для великих наборів даних. Pandas і Matplotlib застосовуються для швидких перевірок і складних графіків.

4. На етапі цільової візуалізації створюються графіки для перевірки гіпотез, використовуючи комбінацію seaborn, Matplotlib і Plotly залежно від завдання. Наприклад, для аналізу впливу категоріальних змінних зручно використати barplot із seaborn із подальшим налаштуванням у Matplotlib.

5. Підсумкове представлення результатів використовує статичні (Matplotlib, seaborn) або інтерактивні (Plotly) графіки для формування звітів чи дашбордів. Plotly інтегрується з фреймворком Dash для створення повноцінних інтерактивних веб-додатків.

Алгоритм методики представлено на рис. 1, де показано послідовність етапів і взаємодію аналітика з інструментами.

Методика була апробована на наборі даних «Титанік», який містить числові (вік, вартість квитка) та категоріальні (стать, клас каюти) змінні, а також пропущені значення [12]. Завдання полягало у виявленні факторів, що впливали на виживання пасажирів.

Етап 1. Швидкий первинний огляд. Дані завантажено в Pandas DataFrame, отримано їх базову статистику та розподіли (рис. 2). Гістограми, згенеровані методом df.hist(), показали, що розподіл віку близький до нормального з піком на молодших пасажирів, а вартість квитка має сильне зміщення вліво.

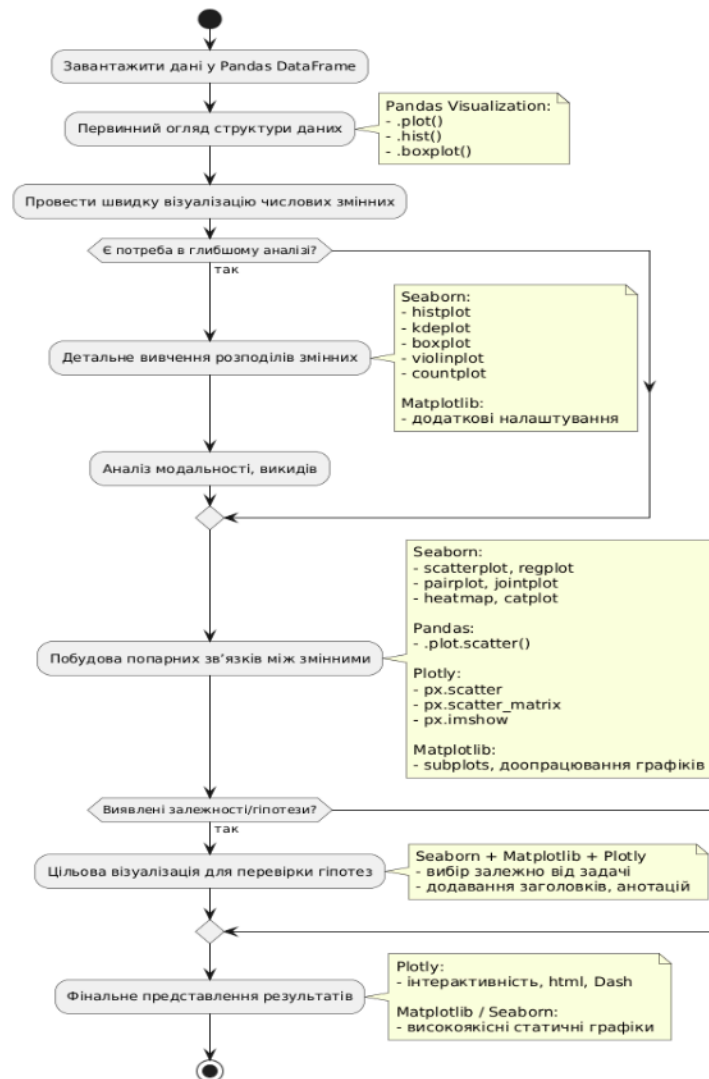


Рис. 1. Етапи розробленої методики візуального аналізу даних

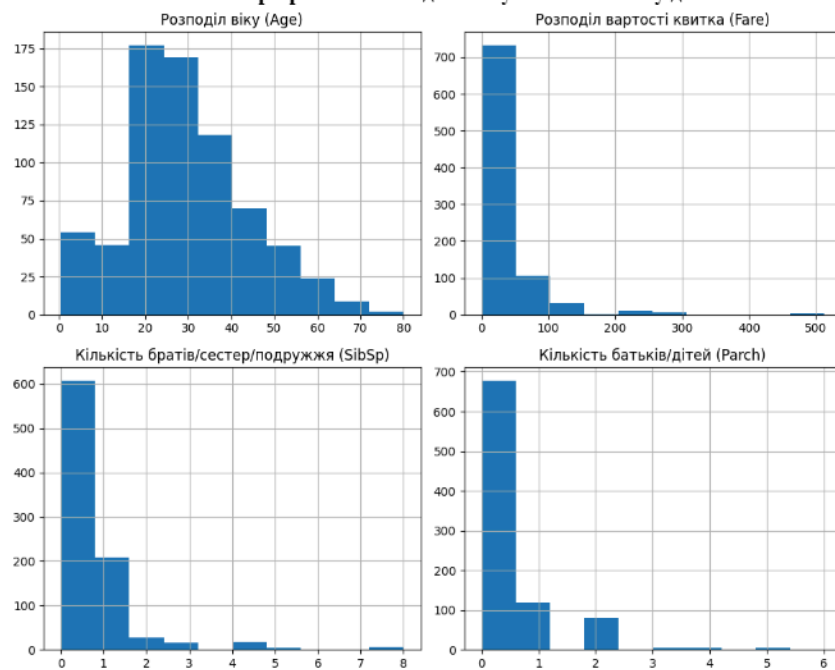


Рис. 2. Гістограми розподілу числових даних «Титанік»

Етап 2. Поглиблений одновимірний аналіз. Бібліотеку seaborn використано для створення деталізованих гістограм із кривими щільності (KDE). Розподіл віку (рис. 3, а) підтвердив піки на дітях і молодих дорослих, а розподіл вартості квитка (рис. 3,б) показав довгий «хвіст» дорогих квитків. Графік типу countplot (рис. 4) виявив, що більшість пасажирів не вижили, переважали чоловіки та пасажирів 3-го класу.

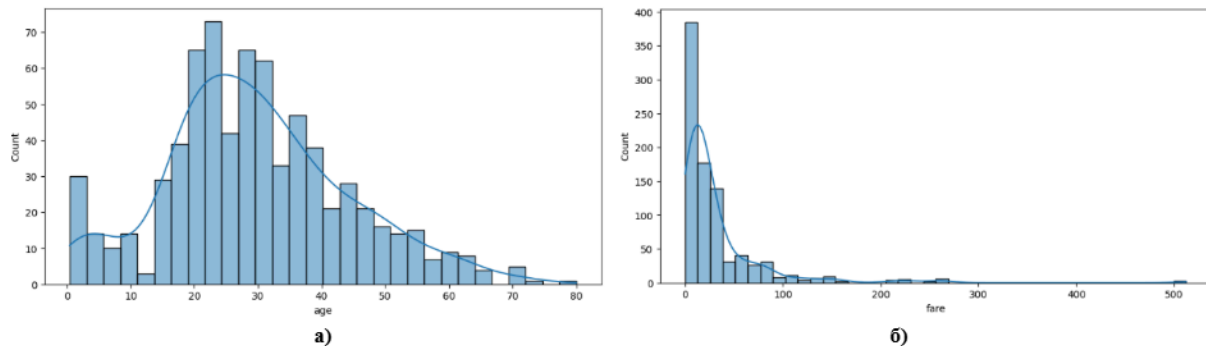


Рис. 3. Деталізовані гістограми віку пасажирів (а) та вартості квитка (б)

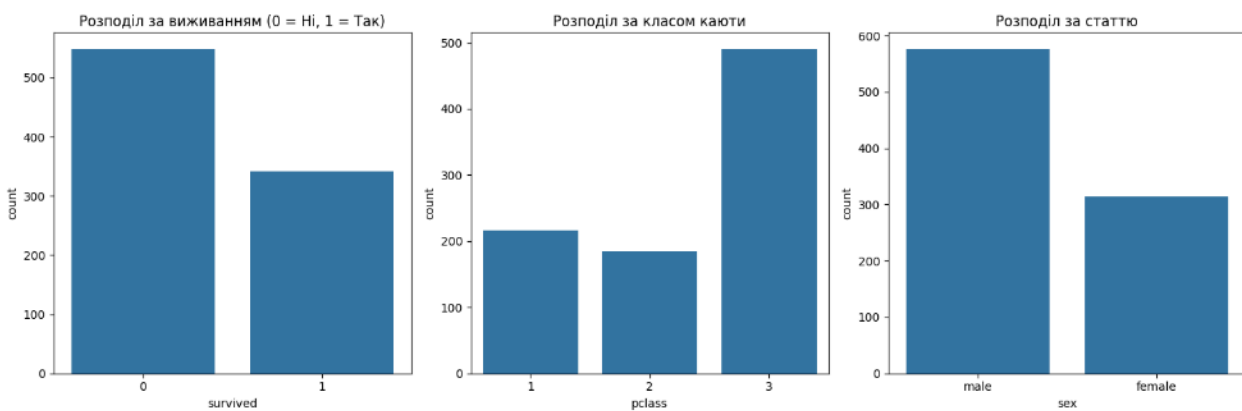


Рис. 4. Розподіл за виживанням, класом і статтю

Етап 3. Двовимірний і багатовимірний аналіз. Бібліотеки seaborn і Plotly застосовано для дослідження взаємозв'язків між даними. Діаграма розсіювання (рис. 6) показала, що виживання частіше асоціюється з вищою вартістю квитка та молодшим віком. Теплова карта (рис. 7) виявила негативну кореляцію між класом каюти та вартістю квитка (-0.55). Коробкові діаграми (рис. 8) підтвердили, що пасажирів 1-го класу були старшими, а діти 2-го і 3-го класів мали вищі шанси вижити. Varplot (рис. 9) показав, що частка жінок, які вижили, значно вища. Інтерактивна діаграма Plotly (рис. 10) дозволила детально дослідити дані через масштабування та фільтрацію.

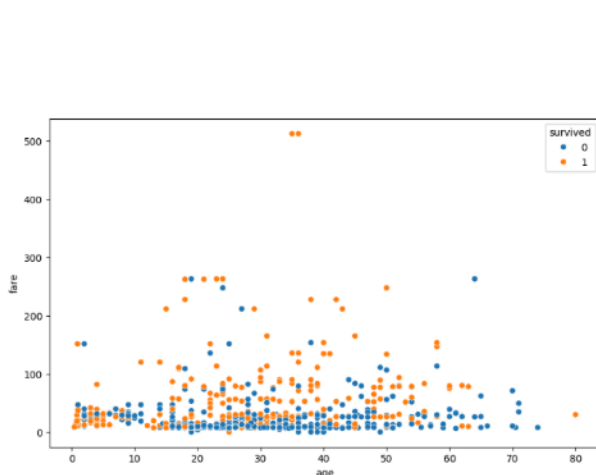


Рис. 6. Взаємозв'язок віку, вартості квитка та виживання

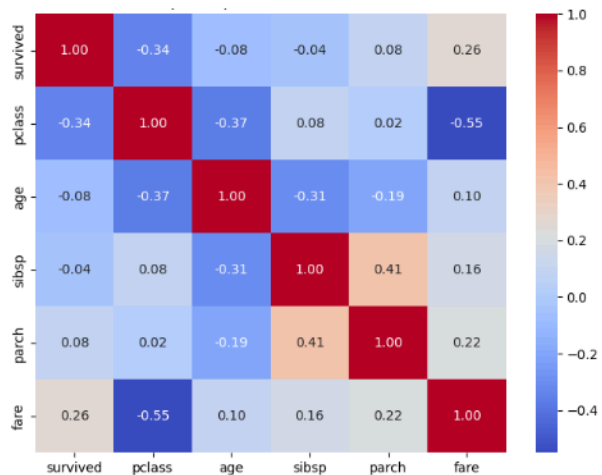


Рис. 7. Теплова карта кореляцій числових змінних

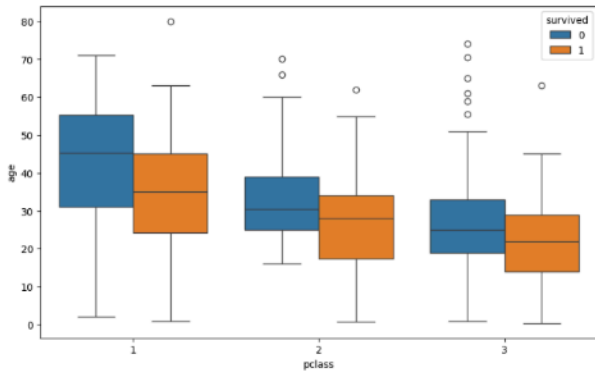


Рис. 8. Розподіл віку за класом і виживанням

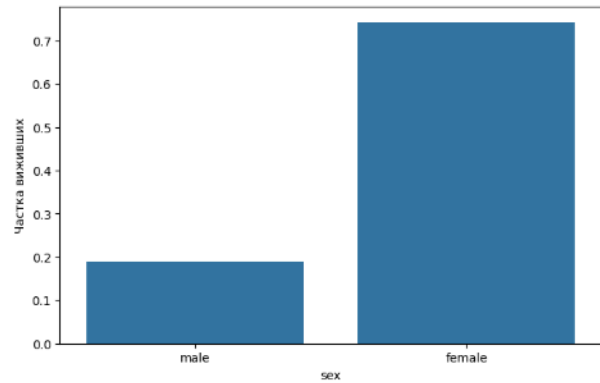


Рис. 9. Частка тих, хто вижив за статтю

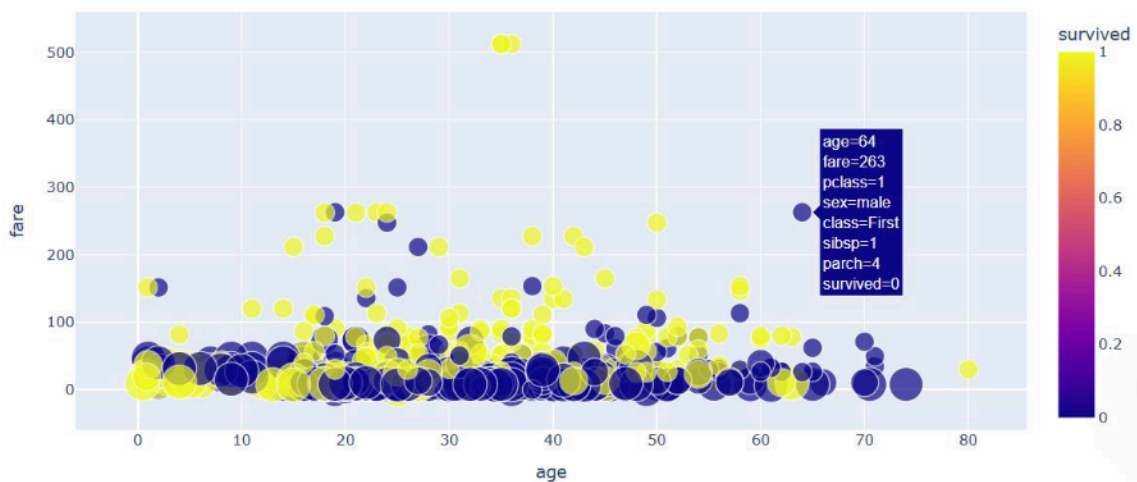


Рис. 10. Інтерактивний аналіз віку, вартості квитка та виживання

Етап 4. Цільова візуалізація. Досліджено вплив класу каюти та статі на виживання за допомогою `seaborn.catplot()` (рис. 11). Жінки мали вищі шанси вижити у всіх класах, особливо в 1-му та 2-му, тоді як чоловіки 3-го класу мали найнижчі шанси.

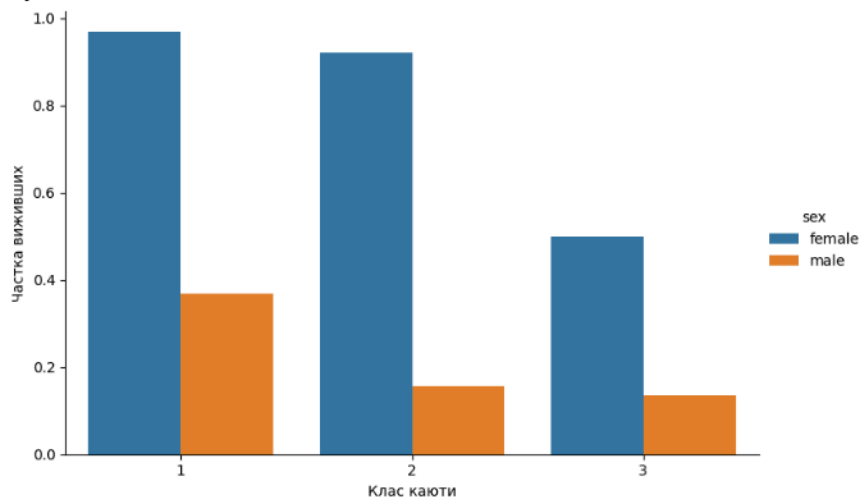


Рис. 11. Вплив класу каюти та статі на виживання

Етап 5. Підсумкове представлення результатів. Інтерактивний графік Plotly (рис. 12) узагальнив фактори виживання, дозволяючи користувачу досліджувати дані через фільтри та спливаючі підказки. Графік підтвердив, що стать, клас і вартість квитка були ключовими факторами.

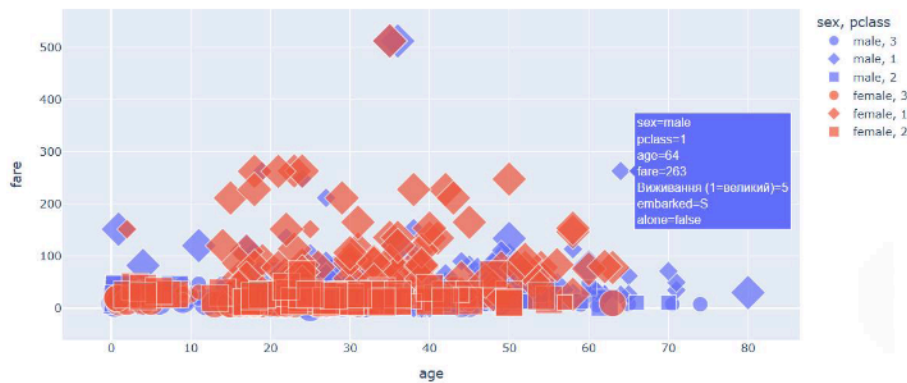


Рис. 12. Підсумковий інтерактивний графік факторів виживання

На основі апробації методики сформульовано рекомендації (табл. 2), які допомагають вибрати бібліотеки залежно від мети, типу даних і аудиторії.

Таблиця 2.

Рекомендації щодо вибору бібліотек Python

Мета	Тип даних	Аудиторія	Рекомендована бібліотека	Тип графіка
Початкове дослідження	Числові, категоріальні	Аналітик	Pandas, seaborn	Гістограми, countplot
Глибокий аналіз	Числові, багатовимірні	Технічні колеги	seaborn, Matplotlib	scatterplot, heatmap, pairplot
Представлення результатів	Усі типи	Нетехнічні	Plotly, seaborn	barplot, інтерактивні діаграми
Дашборди	Часові, числові	Користувачі	Plotly	Лінійні графіки, KPI

Розроблена методика забезпечує структурований підхід до візуального аналізу даних, інтегруючи теоретичні принципи з практичними інструментами Python.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

I ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ У ДАНОМУ НАПРЯМІ

Розроблена методика структурованого підходу до візуального аналізу даних засобами Python продемонструвала ефективність у систематизації процесів обробки інформації. Результати апробації на наборі даних «Титанік» підтвердили, що використання бібліотек Matplotlib, seaborn і Plotly забезпечує гнучкість, відтворюваність і економічну доцільність візуалізацій. Запропоновані етапи – від первинного огляду до підсумкового представлення – дозволяють виявляти закономірності, тенденції та аномалії, сприяючи прийняттю обґрунтованих рішень. Інтеграція інтерактивних інструментів, зокрема Plotly, підвищує доступність результатів для нетехнічної аудиторії, а рекомендації щодо вибору бібліотек оптимізують процес аналізу залежно від типу даних і цілей.

Подальший розвиток може бути спрямований на інтеграцію методики з інструментами машинного навчання для автоматизації вибору типів графіків, розширення підтримки великих даних через оптимізацію бібліотек, таких як HoloViz, та розробку універсальних шаблонів для автоматизованих дашбордів.

Література

1. Matplotlib: Visualization with Python. URL: <https://matplotlib.org/>
2. Seaborn: statistical data visualization. URL: <https://seaborn.pydata.org/>
3. Plotly Open Source Graphing Library for Python. URL: <https://plotly.com/python/>
4. McNutt A. M., Chugh R. Integrated Visualization Editing via Parameterized Declarative Templates. *CHI '21: CHI Conference on Human Factors in Computing Systems*, Yokohama Japan. New York, NY, USA, 2021. DOI: 10.1145/3411764.3445356
5. Assessing the Performance of Python Data Visualization Libraries: A Review / A. Lavanya et al. *International Journal of Computer Engineering in Research Trends*. 2023. Vol. 10, no. 1. P. 28–39. DOI: 10.22362/ijcert/2023/v10/i01/v10i0104
6. A Comparative Analysis of Open-Source Business Intelligence Platforms for Integration with a Low-Code Platform / D. Aveiro et al. *31st International Conference on Information Systems Development*, Lisbon, Portugal, 30 August – 1 September 2023. Lisbon, Portugal, 2023. DOI: 10.62036/isd.2023.2

7. Jorge F.-A., Fernando M., Carlos S.-R. A methodological approach for enhancing visualization of country data representation in the presence of significant spatial disparity. *MethodsX*. 2024. P. 102833. DOI: 10.1016/j.mex.2024.102833
8. Achanta A., Boina R. Advanced Techniques in Python for Effective Data Visualization. *International Journal of Science and Research (IJSR)*. 2023. Vol. 12, no. 12. P. 1919–1926. DOI: 10.21275/sr231227034434
9. The practical value of Seaborn for statistical analysis. URL: <https://medium.com/data-science/seaborn-essentials-for-data-visualization-in-python-291aa117583b>
10. Plotly express for large data sets. URL: <https://stackoverflow.com/questions/73034433/plotly-express-for-large-data-sets>
11. Building Interactive Data Science Applications with Python. URL: <https://www.kdnuggets.com/building-interactive-data-science-applications-with-python>
12. Titanic Data Science Solutions. URL: <https://www.kaggle.com/code/startupsci/titanic-data-science-solutions>

References

1. Matplotlib: Visualization with Python. URL: <https://matplotlib.org/>
2. Seaborn: statistical data visualization. URL: <https://seaborn.pydata.org/>
3. Plotly Open Source Graphing Library for Python. URL: <https://plotly.com/python/>
4. McNutt A. M., Chugh R. Integrated Visualization Editing via Parameterized Declarative Templates. *CHI '21: CHI Conference on Human Factors in Computing Systems*, Yokohama Japan. New York, NY, USA, 2021. DOI: 10.1145/3411764.3445356
5. Assessing the Performance of Python Data Visualization Libraries: A Review / A. Lavanya et al. *International Journal of Computer Engineering in Research Trends*. 2023. Vol. 10, no. 1. P. 28–39. DOI: 10.22362/ijcert/2023/v10/i01/v10i0104
6. A Comparative Analysis of Open-Source Business Intelligence Platforms for Integration with a Low-Code Platform / D. Aveiro et al. *31st International Conference on Information Systems Development*, Lisbon, Portugal, 30 August – 1 September 2023. Lisbon, Portugal, 2023. DOI: 10.62036/isd.2023.2
7. Jorge F.-A., Fernando M., Carlos S.-R. A methodological approach for enhancing visualization of country data representation in the presence of significant spatial disparity. *MethodsX*. 2024. P. 102833. DOI: 10.1016/j.mex.2024.102833
8. Achanta A., Boina R. Advanced Techniques in Python for Effective Data Visualization. *International Journal of Science and Research (IJSR)*. 2023. Vol. 12, no. 12. P. 1919–1926. DOI: 10.21275/sr231227034434
9. The practical value of Seaborn for statistical analysis. URL: <https://medium.com/data-science/seaborn-essentials-for-data-visualization-in-python-291aa117583b>
10. Plotly express for large data sets. URL: <https://stackoverflow.com/questions/73034433/plotly-express-for-large-data-sets>
11. Building Interactive Data Science Applications with Python. URL: <https://www.kdnuggets.com/building-interactive-data-science-applications-with-python>
12. Titanic Data Science Solutions. URL: <https://www.kaggle.com/code/startupsci/titanic-data-science-solutions>