НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Фізико-математичний факультет
Кафедра математичного аналізу та теорії ймовірностей

«На правах рукопису»
УДК 519.237

До захисту допущено:

Завідувач кафедри

Олег КЛЕСОВ

«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою «Страхова та фінансова математика»

зі спеціальності 111 «Математика»

на тему: «Дослідження результатів опитування методами логістичної регресії та факторного аналізу»

Виконала:
студентка 2 курсу магістратури, групи ОМ-31мп
Врубльовська Олена Олександрівна _____

Керівник:
Старший викладач кафедри математичного аналізу та теорії ймовірностей
Кандидат фізико-математичних наук
Мулик Олена Василівна _____

Рецензент:
Професор кафедри статистики, інформаційних технологій та математичних методів в економіці Національної академії статистики, обліку та аудиту, доктор економічних наук, професор
Герасименко Сергій Сергійович

_____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2024 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-математичний факультет
Кафедра математичного аналізу та теорії ймовірностей

Рівень вищої освіти – другий (магістерський)

Спеціальність – 111 «Математика»

Освітньо-наукова програма «Страхова та фінансова математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Олег КЛЕСОВ

« »_____2024 р.

ЗАВДАННЯ
на магістерську дисертацію студентці

Врубльовській Олені Олександрівні

1. Тема дисертації «Дослідження результатів опитування методами логістичної регресії та факторного аналізу», науковий керівник дисертації Мулик Олена Василівна, кандидат фізико-математичних наук, старший викладач кафедри математичного аналізу та теорії ймовірностей, затверджені наказом по університету від «06» листопада 2024р. №4981-с

2. Термін подання студентом дисертації: «14» грудня 2024 року.

3. Об’єкт дослідження: академічна успішність українських студентів у період соціальних та воєнних криз, що мають місце у 2022-2024 роках.

4. Предмет дослідження: вплив та взаємозв'язок між стресовими факторами та академічною успішністю студентів.

5. Перелік завдань, які потрібно розробити :

1) Збір даних про навчальні досягнення та емоційний стан студентів.
2) Збір даних про стан здоров’я пацієнтів, шкідливі звички, хвороби тощо;
3) Огляд літератури щодо методів аналізу освітніх даних та застосування математичних методів у психологічних дослідженнях;
4) Розробка математичних моделей для опису рівня академічної успішності та психоемоційного стану студентів, а також для

знаходження взаємозв'язків між станом здоров'я. та шкідливими звичками, способом життя тощо.

5) Аналіз отриманих результатів дослідження.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: 34 слайди.

7. Орієнтовний перелік публікацій: Врубльовська О.О., Мулик О.В.

Аналіз результатів опитування з використанням stepwise method в моделі логістичної регресії на платформі MS Excel (опитування проводилось серед українських студентів у лютому 2024 р.) *Mathematics in Modern Technical University*

8. Дата видачі завдання 04.09.2024

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Збір даних щодо успішності, проведення опитувань за допомогою Google forms	06.02.2024 – 29.02.2024	Виконано
2	Збір даних щодо успішності, проведення опитувань за допомогою Google forms	01.05.2024 – 31.05.2024	Виконано
3	Опрацювання літератури з тематики магістерської дисертації	04.09.2024 – 16.09.2024	Виконано
4	Ознайомлення з методами обробки педагогічної інформації	17.09.2024 – 26.09.2024	Виконано
5	Вивчення математичних методів у психології	27.09.2024 – 05.10.2024	Виконано
6	Збір даних у пацієнтів за допомогою Google forms	06.10.2024-19.10.2024	Виконано
7	Побудова математичних моделей, розробка алгоритмів для аналізу даних	21.10.2024 – 11.11.2024	Виконано
8	Аналіз результатів та висновків	11.11.2024 – 23.11.2024	Виконано
9	Оформлення магістерської дисертації	24.11.2024 – 08.12.2024	Виконано

Студентка Олена ВРУБЛЬОВСЬКА

Науковий керівник Олена МУЛИК

Реферат

Протягом 2022-2024 років у світі відбулося чимало глобальних криз, таких як пандемія COVID-19, повномасштабна війна в Україні, економічна нестабільність. Події цих років змінили назавжди життя людей і вплинули на життя та здоров'я людей. Як наслідок цих подій, загальний рівень мотивації абітурієнтів, навчальної успішності студентів в Україні понизився і, як приклад, можемо взяти результати Національного мультипредметного тесту (НМТ): у 2023 році не склали тест орієнтовно10 тисяч учнів, а вже в 2024 році - 36 тисяч.

Також існує нерозуміння того, як саме вплинули на здоров'я такі чинники, як наприклад коронавірусна хвороба, зміна місця проживання через ведення бойових дій або вживання певних продуктів харчування.

Отже від подій 2022-2024 років постраждали різноманітні категорії населення. Для кожної групи респондентів було проведено окреме дослідження, для того, щоб визначити, які саме фактори впливають на їх стан.

Перше дослідження присвячено оцінці впливу стресових чинників (факторів) на академічну успішність студентів, що навчаються в НТУУ «КПІ ім. Ігоря Сікорського». Завдяки використанню логістичної регресії та факторного аналізу, було виявлено ключові фактори, які вплинули на навчальні досягнення та на мотивацію студентів.

Друге дослідження зосереджене на тому, щоб проаналізувати дані пацієнтів і пояснити які фактори (вживання певних харчових продуктів, сон, фізична активність, захворювання на Covid, вакцинація, перебування на окупованій території, освіта, наявність роботи, тощо) впливають на виявлені в опитуванні фізичні та психологічні відгуки у респондентів.

Мета та завдання роботи:

Метою роботи є оцінити влив різноманітних факторів, що були зумовлені пандемією та війною, на академічну успішність студентів та на здоров'я пацієнтів у період 2022 – 2024 років.

Завданнями першого дослідження є:

1. Проаналізувати математичні методи та вибрати ті, які будуть ефективними при аналізі даних;
2. Виконати опитування серед студентів;
3. На основі отриманих даних застосувати логістичну регресію та факторний аналіз;
4. Проаналізувати отримані результати і визначити, які стресові фактори мають вплив на успішність та мотивацію.

Завданнями другого дослідження є:

1. Проаналізувати математичні методи та вибрати ті, які будуть ефективними при аналізі даних;
2. Виконати опитування серед пацієнтів;
3. Застосувати факторний аналіз на отриманих даних;
4. Проаналізувати отримані результати і визначити, які стресові фактори і які чинники мають вплив на стан здоров'я.

Об'єкт дослідження:

1. Академічна успішність, мотивація в навчанні та вибір професійної діяльності українських студентів у період 2022-2023 років;
2. Здоров'я пацієнтів, на які вплинули війна та пандемія COVID-19, місце проживання, харчові звички.

Предмет дослідження:

1. Вплив стресових факторів на академічну успішність, мотивацію в навчанні та вибір професійної діяльності українських студентів у період 2022-2023 років;

2. Вплив різноманітних факторів (військові дії, хвороби, харчування тощо) на здоров’я пацієнтів.

Методи дослідження: Логістична регресія, факторний аналіз, анкетування, статистичні характеристики.

Публікації: Врубльовська О.О., Мулик О.В. Аналіз результатів опитування з використанням stepwise method в моделі логістичної регресії на платформі MS Excel (опитування проводилось серед українських студентів у лютому 2024 р.) *Mathematics in Modern Technical University*.

Ключові слова: фактори, академічна успішність, психоемоційний стан, мотивація, здоров’я, логістична регресія, факторний аналіз, прогнозування, гіпотеза.

Abstract

In 2022-2024, the world experienced many global crises, such as the COVID-19 pandemic, the full-scale war in Ukraine, and economic instability. The events of these years have changed people's lives forever and affected people's lives and health. As a result of these events, the overall level of motivation of applicants and academic performance of students in Ukraine has decreased, and we can take the results of the National Multisubject Test (NMT) as an example: in 2023, approximately 10 thousand students failed the test, and in 2024 - 36 thousand.

There is also a lack of understanding of how health was affected by factors such as coronavirus disease, displacement due to the hostilities, or the consumption of certain foods.

Thus, various categories of the population were affected by the events of 2022-2024. A separate study was conducted for each group of respondents to determine which factors affect their situation.

The first study is devoted to assessing the impact of stressors on the academic performance of students studying at NTUU “KPI”. Through the use of logistic regression and factor analysis, the key factors that influenced academic achievement and student motivation were identified.

The second study focuses on analyzing patient data and explaining which factors (consumption of certain foods, sleep, physical activity, Covid disease, vaccination, staying in the occupied territory, education, employment, etc.) influence the physical and psychological responses of respondents identified in the survey.

Purpose and objectives of the study:

The aim of the study is to assess the impact of various factors caused by the pandemic and war on students' academic performance and patient health in the period 2022-2024.

The objectives of the first study are:

1. Analyze mathematical methods and choose those that will be effective in data analysis;
2. To conduct a survey among students;
3. Based on the data obtained, apply logistic regression and factor analysis;
4. Analyze the results and determine which stress factors have an impact on academic performance and motivation.

The objectives of the second study are:

1. Analyze mathematical methods and choose those that will be effective in data analysis;
2. Conduct a survey among patients;
3. Apply factor analysis to the data obtained;
4. Analyze the results and determine which stressors and factors have an impact on health status.

Object of research:

1. Academic performance, motivation in learning and choice of professional activity of Ukrainian students in the period of 2022-2023;
2. The health of patients affected by the war and the COVID-19 pandemic, place of residence, eating habits.

Subject of the study:

1. Influence of stress factors on academic performance, motivation in learning and choice of professional activity of Ukrainian students in the period 2022-2023;
2. The impact of various factors (military operations, illness, nutrition, etc.) on the health of patients.

Research methods: Logistic regression, factor analysis, questionnaires, statistical characteristics.

Publications: Vrublovska O.O., Mulyk O.V. (2024) Analysis of the survey results using the stepwise method in the logistic regression model on the MS Excel platform (the survey was conducted among Ukrainian students in February 2024) Mathematics in Modern Technical University.

Keywords: factors, academic performance, psycho-emotional state, motivation, health, logistic regression, factor analysis, forecasting, hypothesis.

Зміст

Перелік умовних позначень та термінів

Логістична регресія (*англ. logistic regression*) — це статистичний метод, який використовується у тому випадку, коли залежна змінна є бінарною, тобто може набувати лише двох значень.

$$\ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \sum_{k=1}^{n} \beta_k x_k.,$$

Статистика Вальда:

$$\text{Wald} = \left(\frac{\text{b}}{s.\,\text{e.}}\right)^2$$

Коефіцієнти детермінації:

1. **McFadden's** R^2:

$$R_L^2 = 1 - \frac{LL_1}{LL_0}$$

де LL_1 відноситься до повної логарифмічної правдоподібної моделі, а LL_0 відноситься до моделі з меншою кількістю коефіцієнтів.

2. **Cox and Snell's** R^2:

$$R_{CS}^2 = 1 - e^{-\frac{2}{n}(LL_1 - LL_0)}$$

де n – розмір вибірки.

3. **Nagelkerke's** R^2:

$$R_N^2 = \frac{R_{CS}^2}{1 - e^{2LL_0/n}}$$

Статистика J-Юдена

$$\text{Youden's J index} = \text{TPR} + 1 - \text{FPR}$$

Факторний аналіз (*англ. factor analysis*) - це метод, який застосовують для великої кількості змінних, що зводить ці змінні до набагато меншої кількості величин, що будуть незалежними між собою, які і будуть називатися факторами.

Аналіз головних компонентів - це статистичний метод, який використовується для аналізу взаємозв'язків між великою кількістю змінних і для пояснення цих змінних з точки зору меншого числа змінних, званих головними компонентами, з мінімальною втратою інформації.

Якщо $X = (x_i)$ випадковий вектор. Вектор головних компонент $Y = (y_i)$ визначається фомулою:

$$y_i = \sum_{i=1}^{n} \mathrm{B}_{ij} x_i.$$

Тест Кайзера-Майєра-Олкіна (KMO) міра адекватності вибірки (MSA) для змінної x_i визначається за формулою:

$$KMO_j = \frac{\Sigma_{i \neq j} r_{ij}^2}{\Sigma_{i \neq j} r_{ij}^2 + \Sigma_{i \neq j} u_{ij}^2}$$

де кореляційна матриця $R = [p_{ij}]$, а часткова коваріаційна матриця $U = [u_{ij}]$.

Метод обертання факторних навантажень Varimax:

$$V = \frac{1}{k} \sum_{j=1}^{m} [\sum_{i=1}^{k} (\frac{{b_{ij}}^2}{\varphi_i})^2 - (\frac{1}{k} \sum_{i=1}^{k} \frac{{b_{ij}}^2}{\varphi_i})^2]$$

Вступ

Останні роки 2022-2024 принесли світові багато змін, зокрема пандемію Covid-19, повномасштабне вторгнення Росії в Україну, економічну кризу та політичні зміни. Ці події спричинили зміну в психоемоційному стані людей, на їх соціалізацію, стан здоров'я, успішність у навчанні та на вибір професійної діяльності. Результати програми міжнародного оцінювання студентів PISA, свідчить про те, що 2022 рік (рік початку війни) у порівнянні з 2018 роком демонструє зниження успішності українських учнів: рівень математики понизився на 12 балів, із читання на 38 балів, а із природничо-наукових дисциплін на 19 балів. Українські абітурієнти, які пережили декілька років криз, демонструють зниження успішності в загальному рівні знань, а також зниження мотивації. Наприклад, у 2023 році Національний мультипредметний тест не склало близько 10 тисяч школярів, а вже у 2024 році показник досяг 36 тисяч.

Зміна в емоційному стані торкнулась все населення в Україні. Наведемо також ще один приклад, у січні 2024 році компанією Gradus Research company було представлено аналітичний звіт «Психічне здоров'я та ставлення українців до психологічної допомоги під час війни: хвиля 3». Результати цього звіту показали, що 77% українців мають стрес та високий рівень знервованості, а 52% відчувають сильну тривожність та напругу, 47% респондентів відповіли, що мають поганий настрій та мають проблеми із сном, а 42% почувають себе достатньо роздратованими та відчувають часто злість. Варто зауважити, що пацієнти найбільше перебувають у зоні ризику. Погіршення показників здоров'я може бути пов'язано із тим, що присутня недостатня адаптація до нових умов; погіршення фінансового стану людини; вживання недостатньої кількості продуктів або ж вживання неякісних продуктів; обмеження в доступі до медичних послуг; зростання хронічного стресу.

Метою даної роботи є дослідження і вивчення впливу стресових факторів на академічну успішність студентів, а також знайти відповідь на питання: які саме фактори впливають на здоров’я людей. Для цього було прийнято рішення, що варто застосовувати сучасні методи аналізу даних – логістичну регресію та факторний аналіз.

Робота буде складатися із двох досліджень, перше з яких буде розділено на дві частини: побудова моделі логістичної регресії на тих відповідях, що надали в анкетах опитування, що було організовано у лютому 2024 році (в ньому прийняло участь 40 студентів НТУУ «КПІ ім. Ігоря Сікорського»); знаходження основних факторів, які впливають на навчання (на вибірці із відповідями 150 студентів НТУУ «КПІ ім. Ігоря Сікорського») і друге дослідження: проведення факторного аналізу на даних опитування 1101 пацієнту щодо стану їхнього здоров’я, звичок, харчування тощо.

Результат роботи має важливе практичне значення, оскільки він може сприяти розробці рекомендацій для вдосконалення освітніх програм, підвищення якості медичної допомоги, розробка системи харчування для пацієнтів із різними типами захворювань. Такі заходи можуть покращити адаптацію та життя школярів, студентів та пацієнтів, а також посприяти їхньому особистісному розвитку та добробуту.

Розділ 1. Логістична регресія

1.1. Теоретичні основи логістичної регресії

Бінарні змінні використовуються в статистиці для того, щоб змоделювати ймовірність події (P(E)). Це може бути, наприклад, передбачення такої ймовірності: перемоги/поразки; здоров'я/хвороби тощо. Тобто, якщо бінарна змінна $\boldsymbol{Y}$ є результатом певної події E: $\boldsymbol{y_i = 1}$ - означає, що подія відбулася; $\boldsymbol{y_i = 0}$ - подія не відбулася, то метою стає оцінка ймовірності настання події E: $P(Y_i = 1) = p_i$ чи не настання події E: $P(y_i = 0) = 1 - p_i$ на основі попередніх даних або характеристик.

Щоб оцінити таку ймовірність, то найбільш придатною моделлю є **логістична регресія**.

Означення 1. Логістична регресія (*англ. logistic regression*) — це статистичний метод, який використовується у тому випадку, коли залежна змінна є бінарною, тобто може набувати лише двох значень. Основною метою цього регресійного методу є виявлення взаємозв'язків між незалежними змінними та ймовірністю певного результату.

Будується мультиплікативна модель, котра описує залежність шансів результуючої бінарної змінної **Y** від факторних змінних н $X_1, X_2, \ldots X_n$. Формула такої моделі буде виглядати як лінійна комбінація однієї чи декількох незалежних змінних:

$$\ln\big(Odds(E)\big) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon,$$

де $\beta_i, i = \overline{1,n}$ коефіцієнти моделі; ε – випадкова похибка.

Означення 2. $Odds(E)$- це функція, яка вказує шанси, що подія E відбудеться, а саме:

$$Odds(E) = \frac{P(E)}{P(E')} = \frac{P(E)}{1 - P(E)}$$

Якщо p має значення від 0 до 1, тобто є значенням ймовірності, то ми можемо визначити Odds(E), як $Odds(E) = \frac{p}{1-p}$.

Для наших цілей функція шансів (Odds(p)) має перевагу в тому, що перетворює функцію ймовірності, яка має значення від 0 до 1, в еквівалентну функцію зі значенням від 0 до ∞.

Означення 3. Якщо p є ймовірністю, $\frac{p}{1-p}$ – є відповідними шансами, то функція $logit$ визначається як логарифм функції шансів, тобто:

$$logit(E) = lnOdds(E)$$

$$logit(p) = lnOdds(p) = \ln\left(\frac{p}{1-p}\right) = \ln(p) - \ln(1-p)$$

Означення 4. Нехай $\pi = P(E)$, тоді $logit(\pi) = lnOdds(\pi) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$.

Отже, $\frac{P(E)}{1-P(E)} = Odds(E) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$

З цього випливає, що:

$$p = P(E) = \frac{e^{b_0+b_1X_1+b_2X_2+\cdots+b_nX_n}}{1+e^{b_0+b_1X_1+b_2X_2+\cdots+b_nX_n}} = \frac{1}{1+e^{-(b_0+b_1X_1+b_2X_2+\cdots+b_nX_n)}} =$$

$$= \frac{1}{1+e^{-(b_0+\sum_{k=1}^{n} b_k x_k)}}$$

Тут відбувається перехід до моделі, що заснована на спостережуваній вибірці (і тому параметр π замінюється його вибірковою оцінкою p, коефіцієнти β_i — на вибіркові оцінки b_i, а випадкова похибка ε відкидається). Для наших цілей приймаємо E як подію, коли залежна змінна у має значення 1. Якщо у приймає лише значення 0 або 1, то можемо розглядати E як успіх, а E' як невдачу.

Нехай E_i- це подія, що $\boldsymbol{y_i = 1}$ **та** $p_i = P(E_i)$, подібно до того, як лінія регресії дає можливість передбачити значення залежної змінної у за

значеннями незалежних змінних $x_1, x_2, \ldots x_n$, для логістичної регресії маємо:

$$p = P(y = 1) = \frac{1}{1 + e^{-(b_0 + \sum_{j=1}^{k} b_j x_j)}}$$

$$logit(p) = ln\frac{p}{1-p} = \ln\left(\frac{P(y=1)}{1 - P(y=1)}\right) = b_0 + \sum_{j=1}^{k} b_j x_j.$$

За умови, коли $k = 1$:

$$p = \frac{1}{1 + e^{-(b_0 - b_1 x)}}$$

Така крива буде мати сигмоподібну форму, як показано на рис. 1.

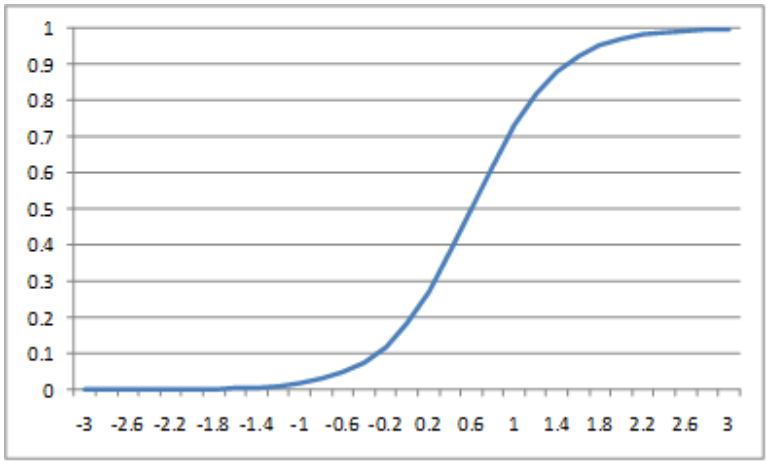


Рис. 1 Сигмоїда для p

Важливим моментом буде введення функції «відношення шансів» (Odds Ratio OR, $R_{x_i x_j}$), яка показує в якому відношенні ймовірність одного випадку буде більше/менше, ніж другого при всіх інших однакових умовах:

$$R_{x_i x_j} = \frac{Odds(x_{i1}, \cdots, x_{in})}{Odds(x_{j1}, \cdots, x_{jn})} = \frac{e^{\beta_0 + \sum_{k=1}^{n} \beta_k x_{ik}}}{e^{\beta_0 + \sum_{k=1}^{n} \beta_k x_{jk}}} = e^{\sum_{k=1}^{n} \beta_k (x_{ik} - x_{jk})}.$$

Якщо записати через функцію logit для випадку коли $n = 1$ та $P(y = 1) = \frac{1}{1+e^{\beta_0 + \beta_1 x_1}}$ то отримаємо наступну формулу:

$$logit\left(\frac{p_{x+1}}{p_x}\right) = \ln\left(\frac{p_{x+1}}{1-p_{x+1}}\Big/\frac{p_x}{1-p_x}\right) = \beta_0 + \beta_1(x+1) - \beta_0 + \beta_1 x = \beta_1,$$

Тоді:

$$\frac{Odds(x+1)}{Odds(x)} = \frac{p_{x+1}}{1-p_{x+1}}\Big/\frac{p_x}{1-p_x} = e^{\beta_1},$$

тобто, якщо є $x = 1$ та $x = 0$, тоді величина e^{β_1} представляє відношення шансів між цими значеннями.

Якщо маємо подію E_i, тоді її ймовірність буде тим вище, чим більше значення $Odds(E_i)$:

$$Odds(E_i) = \frac{event}{not\ event}.$$

Для логістичної моделі не можна використовувати метод найменших квадратів для розрахунку значень коефіцієнтів $b_i, i = \overline{1,n}$. Натомість обчислення проводиться методом максимальної правдоподібності. Модель, що використовується, заснована на біноміальному розподілі, функція щільності ймовірності $f(x;\theta) = C_n^k p^k (1-p)^{n-k}$ події для вибіркових даних визначається формулою:

$$L = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

де y_i — спостережувані значення, а p_i — відповідні теоретичні значення.

Означення 5. Логарифмічна функція правдоподібності – це функція, що вимірює наскільки добре статистична модель пояснює спостережувані дані і записується за наступною формулою:

$$LL = \ln L = \sum_{i=1}^{n}(y_i \ln p_i + (1-y_i)\ln(1-p_i))$$

Метою є максимізувати функцію LL, припускаючи, що $p_i = \frac{1}{1+e^{-(b_0+\sum_{k=1}^{n} b_k x_k)}}$, для отримання найкращих значень b_i. Даний процес

можна зробити вручну застосовуючи Solver (Data Analysis). Також обчислюється значення функції LL_0, яка є початковим значенням для LL, оскільки враховує лише значення коефіцієнту перетину (intercept b_0) і яку можна обрахувати за наступною формулою.

$$LL_0 = \ln L_0 = n_0 \ln \frac{n_0}{n} + n_1 \ln \frac{n_1}{n},$$

де n_0 – кількість спостережень із значенням 0; n_1- кількість спостережень зі значенням 1 і $n = n_0 + n_1$.

Пакет ресурсів Real Statistics надає інструмент аналізу даних Logistic and Probit Regression. Дана платформа приймає як вхідні дані діапазон, що містить вибіркові незалежні дані у вигляді стовпців $X_1, X_2, \ldots X_n$ та стовпець з бінарною результуючою змінною $\boldsymbol{Y}$ успішних і невдалих випадків.

1.2. Побудова моделі. Використання платформи MS Excel (Real Statistics) для обчислень.

У лютому 2024 року було проведено анонімне опитування студентів 2-3 курсів (N=40), які навчаються в НТУУ «КПІ ім. Ігоря Сікорського» із метою виявлення факторів, які впливають на їх мотивацію, бажання продовжити професійну діяльність в Україні. Для аналізу було відібрано 11 запитань, що оцінювалися за шкалою від 1 до 5. На рис.2 зображена частина вихідної статистики, котра використовувалась у цьому дослідженні.

X1	X2	X4	X6	X7	X8	X9	X10	X11	X12	Y
5	2	2	4	5	5	4	3	3	4	1
5	3	2	5	5	5	5	4	5	3	1
4	1	1	2	5	4	3	4	3	2	1
5	5	1	5	5	5	5	5	1	2	0
5	3	5	5	5	3	1	4	3	4	0
4	1	2	3	4	4	3	2	4	4	1
4	2	1	4	5	4	4	5	5	4	1
5	2	4	3	4	2	2	3	3	4	1
4	2	4	4	5	2	4	1	2	3	0
2	4	4	2	5	2	1	1	3	3	1
4	4	3	4	4	2	2	3	4	3	0
5	1	1	5	5	5	5	5	1	2	0
5	2	3	5	5	5	2	4	3	3	0
5	3	3	4	5	5	5	5	1	2	0
4	4	3	4	4	4	4	3	3	2	0
5	3	1	4	5	4	2	1	2	4	0
4	2	1	4	5	3	4	5	3	3	0
3	2	1	4	3	2	3	4	3	3	1
4	3	1	4	4	5	4	3	3	1	1
4	4	1	5	4	3	4	4	2	1	0
4	5	5	3	5	2	4	5	3	4	1
2	1	1	2	2	2	2	1	3	4	1
3	1	1	1	3	2	2	1	4	4	0
3	4	4	3	4	4	3	4	2	3	0
4	5	2	5	5	5	4	1	1	2	0
4	3	5	4	5	4	4	4	2	4	0
5	2	4	3	5	4	3	4	3	4	0
5	4	2	3	5	4	3	4	3	4	0

Рис.2 Частина вихідної статистики: вибіркові незалежні дані у вигляді стовпців $X_1, X_2, \ldots X_{12}$; бінарна результуюча змінна ***Y***

Для моделювання залежностей було використано логістичну регресію**,** котра була реалізована за допомогою платформи MS Excel (модуль *Real Statistics*). Вихідна статистика для першого кроку представлена на Рис. 3.

Logistic Regression							
		# Iter	**20**		**Alpha**	**0,05**	
	coeff	**s.e.**	**Wald**	**p-value**	**exp(b)**	**lower**	**upper**
intercept	-0,92712	4,089956	0,051384	0,820672	0,395694		
X1	-1,65638	0,79661	4,323451	0,037591	0,190828	0,040047	0,909321
X2	-0,74176	0,511647	2,101797	0,147126	0,476273	0,174719	1,298287
X4	0,207677	0,454471	0,208816	0,647697	1,230816	0,505064	2,999438
X6	-0,50152	0,547398	0,839399	0,359569	0,60561	0,207132	1,770677
X7	0,438482	0,70124	0,390993	0,531778	1,550351	0,392223	6,128115
X8	0,160132	0,521693	0,094216	0,758885	1,173665	0,422161	3,262951
X9	0,927906	0,561646	2,729504	0,09851	2,529208	0,841223	7,604281
X10	0,279727	0,421491	0,440447	0,506907	1,322769	0,579043	3,021745
X11	1,329988	0,719014	3,421532	0,064351	3,780997	0,923805	15,47505
X12	0,004539	0,596538	5,79E-05	0,993929	1,00455	0,312031	3,234038

Рис. 3 Частина вихідної статистики: обчислені коефіцієнти моделі $b_0, b_1, b_2, \ldots b_{12}$.

В результаті проведених розрахунків отримуємо в таблиці **Logistic Regression** (рис.3):

1. Значення **coeff** b_i – коефіцієнти першої моделі логістичної регресії;
2. Стандартна похибка (***s.e***);
3. Статистика Вальда (**Wald) –** показує, наскільки значущим є фактор і визначається за формулою:

$$\text{Wald} = \left(\frac{\text{b}}{s.\,\text{e.}}\right)^2$$

Ця статистика є приблизно нормально розподіленою і використовується для перевірки відмінності від 0 для кожного i-го коефіцієнту моделі і відповідає на питання, чи впливає дана i-та факторна ознака (з урахуванням стандартизації за всіма іншими факторами) на вихідну змінну. Чим більше величина $\boldsymbol{Wald}$, тим значніше впливає ця ознака на результат, статистика Вальда приблизно повторює розподіл $\chi^2(df) \sim (Wald)^2$.

4. **P-value** показує рівень значущості впливу кожного фактора. Розраховується як: $= CHIDIST(Wald; df)$, де $df = 1$ (CHISQ.DIST.RT). Саме на стовпець **p-value** потрібно орієнтуватись для визначення мінімального набору факторних ознак, пов'язаних з вихідною змінною використовується метод покрокового відкидання/ включення змінних (Stepwise), тобто виключаємо факторну ознаку з найвищим значенням p-value.
5. Останні три стовці є показником відношенням шансів (ВШ, **odds ratio**):

   5.1. **Exp(b);**

   5.2. **Lower**, **upper** – межі довірчого інтервалу для ВШ, що обчислюються за формулою:

$$EXP(b \pm Se * NORMSINV\left(1 - \frac{\alpha}{2}\right)$$

Значення функції відношення шансів (Odds Ratio) більш точно оцінюють ступень впливу кожної факторної змінної в логістичній моделі регресії (з урахуванням стандартизації за всіма іншими факторами) на вихідну змінну. Якщо значення показника i-ої факторної ознаки ВШ > 1 при $b_i > 0, (p - value < 0{,}05)$, то можна сподіватись на зростання шансів випадку при зростанні цієї ознаки, якщо ВШ<1 при $b_i < 0, (p - value < 0{,}05)$ - це свідчить про зниження шансів випадку при зростанні цієї ознаки. Якщо ж значення показника ВШ i-ої факторної ознаки статистично значимо не відрізняється від 1, то зміна цієї ознаки не пов'язана зі зміною ризику випадку.

1.3. Перевірка відповідності моделі логістичної регресії

1.3.1. Вдосконалення перевірки значущості

На жаль, для більшості значень коефіцієнта логістичної регресії b стандартна помилка та пов'язана із нею статистика Вальда є завищеними. Це збільшує ймовірність того, що b буде вважатися таким, що не робить суттєвого внеску в модель, навіть якщо такий внесок існує (тобто помилка II роду).

Щоб подолати цю проблему, краще виконати перевірку значущості за допомогою статистики логарифмічної ймовірності, тобто:

$$2(LL_1 - LL_0) \sim (-2LL_0) - (-2LL_1) \sim \chi^2(df),$$

де $df = k - k_0$ - кількість факторів у повній моделі мінус виключені фактори.; LL_1 – відноситься до повної логарифмічної моделі правдоподібності; LL_0 – відноситься до моделі із меншою кількістю коефіцієнтів.

Тобто, це еквівалентно

$$-2ln\frac{L_0}{L_1}\sim\chi^2(df)$$

Застосуємо даний тест для вибірки:

$$LL_0 = \ln L_0 = 26\ln\frac{26}{40} + 14\ln\frac{14}{40} = -25{,}8979$$

Значення LL обчислюється Logistic and Probit Regression на платформі MS Excel і становить $-16{,}89$. Для обчислення χ^2 використовуються формули: $\chi^2(10) = 2(LL - LL_0) = 18{,}003$ і p-value$= \text{CHISQ.DIST.RT}(18{,}003; 10) = 0{,}0549$.

На рис. 4 в стовпці «LL statistics» можемо побачити результати

Coeff
-0,92712
-1,65638
-0,74176
0,20768
-0,50152
0,43848
0,16013
0,92791
0,27973
1,32999
0,00454

LL statistics	
LL0	-25,8979
LL1	-16,8962
Chi-Sq	18,0032
df	10
p-value	0,05491
alpha	0,05
sig	no
R-Sq (L)	0,34758
R-Sq (CS)	0,36242
R-Sq (N)	0,49916
AIC	55,7925
BIC	74,3702

Рис. 4 Частина вихідної статистики: вибіркові незалежні дані у вигляді стовпців передбачена ймовірність p_i; «LL statistics»; коефіцієнти та характеристики моделі.

Але даний результат в нашому випадку не є значущим оскільки $p > 0.05$ і потрібно використати метод покрокового включення/виключення факторних змінних (stepwise method) для зменшення їх кількості, що, можливо, покращить модель.

1.3.2. Коефіцієнт детермінації

Вважається, що для звичайної регресії коефіцієнт детермінації має наступний вигляд:

$$R^2 = 1 - \frac{RSS}{TSS}$$

де RSS - сума квадратів залишків регресі; TSS - загальна сума квадратів. Тобто коефіцієнт детермінації вимірює який відсоток дисперсії, пояснено регресійною моделлю, наскільки отримані спостереження підтверджують модель.

1.3.3. Застосування статистик до логістичної регресії.

У нашому дослідженні для логістичної регресії необхідна подібна статистика, як було наведено вище для лінійної регресії, щоб оцінити оцінку якості моделі. Розглянемо три псевдо-коефіцієнти детермінації:

1. Коефіцієнт детермінації **McFadden's** R^2:

$$R_L^2 = 1 - \frac{LL_1}{LL_0}$$

де LL_1 відноситься до повної логарифмічної правдоподібної моделі, а LL_0 відноситься до моделі з меншою кількістю коефіцієнтів.

2. Коефіцієнт детермінації **Cox and Snell's** R^2:

$$R_{CS}^2 = 1 - e^{-\frac{2}{n}(LL_1 - LL_0)}$$

де n – розмір вибірки.

3. Коефіцієнт детермінації **Nagelkerke's** R^2:

$$R_N^2 = \frac{R_{CS}^2}{1 - e^{2LL_0/n}}$$

На рис. 4 в стовпці «LL statistics» можемо побачити розраховані коефіцієнти.

Для оцінки значення коефіцієнтів детермінації використовуються такі емпіричні правила:

- $R^2 < 0.2$: слабкий зв’язок;
- $0.2 \leq R^2 < 0.4$: помірний зв’язок;
- $R^2 \geq 0.4$: сильний зв’язок.

В нашому прикладі значення (рис. 4) $R_L^2 = 0{,}348, R_{CS}^2 = 0{,}362$, що демонструють нам помірний зв'язок , значення $R_N^2 = 0{,}499$ – сильний зв'язок. Отримані дані підтверджують, що побудована нами модель логістичної регресії має достатньо добру пояснювальну здатність і враховує основні залежності у даних.

1.4. Порогове значення (cutoff) у логістичній регресії

Для проведення подальшого аналізу за допомогою логістичної регресії, є надзвичайно вважливим введення такого поняття, як **порогове значення (cutoff).** Застосування порогового значення дозволить класифікувати результати моделі. У нашому випадку, використаємо спочатку стандартне порогове значення $cutoff = 0{,}5$ (проте це значення можна змінювати зілежно від цілей аналізу).

На рис. 5 можемо побачити прогнозовані значення (*p-Pred*) і їхнє значення відносно $cutoff = 0{,}5$ можемо класифікувати, як успішне або неуспішне. Тобто,

- Якщо значення у стовпці $p - Pred \geq 0{,}5$, то результат класифікується як позитивний (успішний), тобто $Y = 1$.
- Якщо значення $p - Pred < 0{,}5$, то результат класифікується як негативний (неуспішний), тобто $Y = 0$.

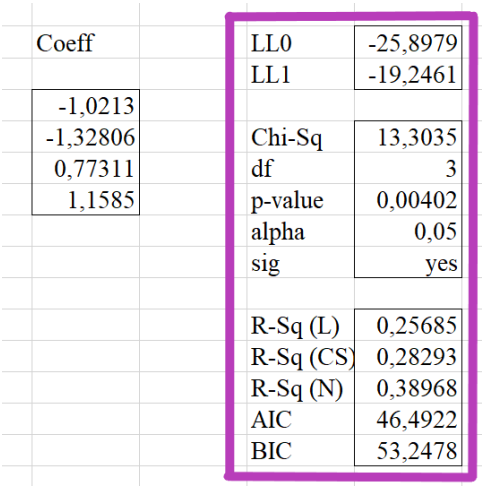
Стовпець *Suc-Pred* – в даному випадку буде повторювати стовпець *p-Pred*, тому що метою було знаходження прогнозованих ймовірностей, відповідних до значень $\boldsymbol{Y = 1}$ та $\boldsymbol{Y = 0}$.

Success	*Failure*	*Total*	*p-Obs*	*p-Pred*	*Suc-Pred*	*Fail-Pred*	*LL*	*% Correct*
1	0	1	1	0,32491	0,32491	0,67509	-1,1242	0
1	0	1	1	0,86858	0,86858	0,13142	-0,14089	100
1	0	1	1	0,83819	0,83819	0,16181	-0,17651	100
0	1	1	0	0,00779	0,00779	0,99221	-0,00782	100
0	1	1	0	0,01513	0,01513	0,98487	-0,01525	100
1	0	1	1	0,8445	0,8445	0,1555	-0,16901	100
1	0	1	1	0,97761	0,97761	0,02239	-0,02264	100
1	0	1	1	0,06984	0,06984	0,93016	-2,66151	0
0	1	1	0	0,26232	0,26232	0,73768	-0,30425	100
1	0	1	1	0,58531	0,58531	0,41469	-0,53561	100
0	1	1	0	0,14185	0,14185	0,85815	-0,15298	100
0	1	1	0	0,13234	0,13234	0,86766	-0,14196	100
0	1	1	0	0,06877	0,06877	0,93123	-0,07125	100
0	1	1	0	0,07965	0,07965	0,92035	-0,083	100
0	1	1	0	0,27719	0,27719	0,72281	-0,32461	100
0	1	1	0	0,00374	0,00374	0,99626	-0,00374	100
0	1	1	0	0,72153	0,72153	0,27847	-1,27845	0

Рис. 5 Частина вихідної статистики

1.5. Метод Stepwise selection

Метод **Stepwise selection (покроковий відбір)** використовується для того, щоб знайти оптимальний набір змінних для моделі логістичної регресії. Мета цього методу полягає у відшуканні баланс між максимальним спрощенням моделі і її якістю (пояснювальною здатністю). Такий покроковий відбір може базуватися на статистичних критеріях, таких як AIC та BIC, та в нашому випадку використаємо $p - value$.



Coeff
-1,0213
-1,32806
0,77311
1,1585

LL0	-25,8979
LL1	-19,2461
Chi-Sq	13,3035
df	3
p-value	0,00402
alpha	0,05
sig	yes
R-Sq (L)	0,25685
R-Sq (CS)	0,28293
R-Sq (N)	0,38968
AIC	46,4922
BIC	53,2478

Рис.6 Модель логістичної регресії після застосування метод Stepwise selection

На рис. 6 зображена нова модель логістичної регресії, побудовану за допомогою методу покрокового відбору. Варто зазначити, що характеристики цієї регресії, демонструють значне покращення порівняно із початковою регресійною моделлю без покрокового відбору (рис.4). Зокрема, значення $p - value$ менше за 0.05, коефіцієнти детермінації свідчать про хорошу якість моделі, а також зменшені значення AIC та BIC свідчать про підвищену ефективність моделі. Нова модель буде використана для подальших розрахунків.

1.6. Оцінка прогностичних характеристик моделі

На Рис. 7 представлена таблиця «Classification Table» яка є іншим способом оцінки якості моделі. Інструмент аналізу даних Real Statistics Logistic Regression створює її на основі заданих формул.

Classification Table

	Obs Suc	Obs Fail	Total
Pred Suc	10	2	12
Pred Fail	4	24	28
Total	14	26	40
Accuracy	0.714286	0.923077	0.85
	Sensitivity	Specificity	
Cutoff	0.5		
AUC	0.876374		

Рис. 7 Прогностичні характеристики логістичної регресії

На рис. 7 показано порівняння кількість успіхів ($\boldsymbol{Y = 1}$), що були передбачені моделлю логістичної регресії, у порівнянні із фактично спостережуваним числом, а також кількість невдач ($\boldsymbol{Y = 0}$), що також були передбачені моделлю логістичної регресії у порівнянні із фактично спостережуваним числом.

Елементами таблиці є чотири можливі результати:

1. **TP (True Positive)** — кількість випадків, які були правильно класифіковані як позитивні, тобто передбачалися як успішні та фактично спостерігалися як успішні;
2. **FP (False Positive)** — кількість випадків, які були неправильно класифіковані як позитивні, тобто передбачалися як успішні, але насправді спостерігалися як невдалі;
3. **TN (True Negative)**— кількість випадків, які були правильно класифіковані як негативні, тобто були передбачені як невдалі та фактично спостерігалися як невдалі;
4. **FN (False Negative)** — кількість випадків, які були неправильно класифіковані як негативні, тобто були передбачені як невдалі, але насправді спостерігалися як успішні.

Тобто «Classification Table» на рис. 7 можна представити у наступному вигляді:

Стан			
	Позитивний стан	**Негативний стан**	
Позитивно прогнозований стан	істино позитивний **TP**	хибно позитивний **FP**	**TP+FP**
Негативно прогнозований стан	хибно негативний **FN**	істино негативний **TN**	**TN+ FN**
	TP+FN	**TN+FP**	**N= TP+FN+ FP+TN**

Рис. 8 Таблиця оцінки прогностичних характеристик моделі

Загальна кількість спостережень у таблиці визначається як:

$$N = TP + FN + FP + TN$$

Також слід зауважити, що FP – це помилка I роду, а FN – II роду.

Якість біостатистичного бінарного тестування (і діагностичного, і скринінгового) характеризується певним балансом між двома важливими статистичними мірами продуктивності моделі – чутливістю та специфічністю. Розглянемо на нашому прикладі обчислення даних характеристик і які саме з них обчислює Real Statistics Logistic Regression:

Чутливість ($\boldsymbol{sensitivity}$) $=$ **True positive rate** (**TRP**) – показує частку істинно позитивних об'єктів ($\boldsymbol{Y = 1}$), які визначені правильно:

$$Sensitivity = \frac{TP}{TP + FN} = \frac{10}{10 + 4} = 0{,}714 \Longrightarrow 71{,}4\%$$

Специфічність ($\boldsymbol{specificity}$) = **True negative rate** (**TNR**)- показує частку істинно негативних результатів ($\boldsymbol{Y = 0}$), серед усіх фактичних негативних випадків:

$$Specificity = \frac{TN}{TN + FP} = \frac{24}{24 + 2} = 0{,}92 \Longrightarrow 92\%.$$

Загальна точність моделі ($\boldsymbol{Accuracy}$) визначається як:

$$Accuracy = \frac{TP + TN}{N} = \frac{10 + 24}{40} = 0{,}85 \Longrightarrow 85\%.$$

На основі отриманих характеристик можемо зробити висновок, що представлена модель є достатньо ефективною, оскільки її точність «Accuracy» становить 85%, що вказує на невеликий рівень похибки - 15%. Отримані результати дозволяють із задовільною точністю прогнозувати реальний стан подій. Однак, модель можна покращити, якщо здатність визначити точно людей, які мають певний досліджуваний стан зросте, а кількість хибнопозитивних та хибнонегативних результатів буде залишатися мінімальною.

Діагностика має на меті визначити максимально точне ймовірнісне визначення осіб, які дійсно мали певний досліджуваний стан. Якщо ж модель буде надавати велику кількість хибнопозитивних або хибнонегативних результатів, то це може спричинити помилки не тільки у прогнозах, а й мати реальні наслідки. Тобто, хибнопозитивні результати можуть вимагати певних додаткових тестувань, а це в свою чергу, може спричиняти додаткову збільшену вартість обстеження, а також психологічний стрес для пацієнта. Хибнонегативні результати можуть нести в собі ризик пропустити проблему, що може призвести до доволі серйозних реальних наслідків (наприклад, погіршення здоров’я,

економічні втрати тощо). Отже, ключовим завданням є зменшення кількості хибних результатів для того, щоб підвищити якість прогнозів та ефективності моделі.

Додатково до чутливості та специфічності, було розраховано прогностичні показники:

1. **Прогностична цінність позитивного результату** *(Positive Predictive Value, PPV):*

$$PPV = \frac{Sensitivity \times Prevalence}{Sensitivity \times Prevalence + (1 - Specificity) \times (1 - Prevalence)} \cdot 100\%$$

$$PPV = \frac{0{,}714 \cdot 0{,}252}{0{,}714 \cdot 0{,}252 + (1 - 0{,}92) \cdot (1 - 0{,}252)} \cdot 100\% = 75{,}04\%$$

Тобто 75,04% усіх випадків, які модель визначає як позитивні, насправді є істинно позитивними.

2. **Прогностична цінність негативного результату** *(Negative Predictive Value, NPV):*

$$NPV = \frac{Specificity \times (1 - Prevalence)}{(1 - Sensitivity) \times Prevalence + Specificity \times (1 - Prevalence)} \cdot 100\%$$

$$NPV = \frac{0{,}92 \cdot (1 - 0{,}252)}{(1 - 0{,}714) \cdot 0{,}252 + 0{,}92 \cdot (1 - 0.252)} \cdot 100\% = 90{,}5\%$$

Отже, 90,5% - це кількість вірних прогнозів, для яких модель прогнозувала відсутність випадку.

3. **Поріг поширеності** *(Prevalence):*

$$Prevalence = \frac{\sqrt{Specificity \times (1 - Sensitivity)} + Specificity - 1}{Sensitivity + Specificity - 1} =$$

$$= \frac{\sqrt{0{,}714(1 - 0{,}92)} + 0{,}92 - 1}{0{,}714 + 0{,}92 - 1} = \frac{0{,}159}{0{,}634} = 0{,}252.$$

Даний результат означає, що 25,2% всіх спостережень у вибірці належать до успіху.

Ці показники демонструють, що модель є більш точною в прогнозуванні негативних результатів, ніж позитивних. Це може бути пояснено специфікою вибірки та сильним впливом факторів, що перешкоджають успішності студентів у складних умовах.

4. **Коефіцієнт невлучання** *(Miss Rate)* - це показник, який характеризує частку істинно позитивних випадків, які не були виявлені тестом, тобто частку хибно негативних результатів серед усіх дійсно позитивних випадків

$$MissRate = \frac{FN}{TP + FN} = \frac{4}{14} = 0.2857 \Rightarrow 28{,}57\%$$

Тобто приблизно 29% позитивних результатів не були розпізнані моделлю.

5. **Хибнопозитивний рівень** (*False Positive Rate*) — це показник, який характеризує частку хибно позитивних результатів серед усіх дійсно негативних випадків.

$$FalsePositiveRate = \frac{FP}{FP + TN} = \frac{2}{26} = 0.0769 \Rightarrow 7.69\%$$

Для нашої моделі він склав 7.69%, що свідчить про невелику кількість хибних прогнозів позитивного результату.

6. **Рівень хибного виявлення** (*False Discovery Rate*) — це показник, який показує частку хибно негативних результатів серед усіх випадків, які були визначені як позитивні.

$$FalseDiscoveryRate = \frac{FN}{FN + TP} = \frac{4}{14} = 0.2857 \Rightarrow 28.5\%$$

У нашому випадку рівень хибного виявлення становить 28.57%. Це означає, що приблизно 28.57% всіх випадків, які були визначені моделлю як позитивні, насправді є хибними.

7. **Рівень хибного пропускання** (*False Omission Rate*) — це показник, який характеризує частку хибно негативних результатів серед усіх випадків, які були визначені як негативні.

$$FalseOmissionRate = \frac{FN}{FN + TN} = \frac{4}{28} = 0.1429 \Rightarrow 14.29\%$$

Це означає, що приблизно 14.29% всіх випадків, які були визначені моделлю як негативні, насправді є хибними.

Аналіз прогностичних характеристик моделі дозволяє оцінити її точність та надійність у прогнозуванні результатів. Хоча модель демонструє високу специфічність (92%) та задовільну чутливість (71.4%), все ж таки існує певна кількість хибнопозитивних та хибнонегативних випадків, що свідчить про можливість подальшого удосконалення моделі.

Регулювання порогового значення $cutoff$ може допомогти покращити баланс між чутливістю та специфічністю залежно від конкретних цілей дослідження.

1.7. Аналіз кривої операційних характеристик (ROC curve analysis)

Прогностичні характеристики логістичної моделі регресії (або деякого тесту) добре описуються кривою операційних характеристик моделі (Receiver Operating Characteristic – ROC curve analysis). ROC-крива – це діаграма, що відображає значення частоти хибнопозитивних результатів (FPR), які зображуються по осі x, в порівнянні із частотою істинно позитивних результатів (TRP), які зображуються по осі y, для всіх можливих граничних значень від 0 до 1. Ідеальна модель матиме значення FRP та TRP рівними 100%, і її крива проходитиме через верхній лівий кут графіку (площа під кривою, AUC = 1).

Інструмент Real Statistics Logistic Regression в MS Excel автоматично створює ROC-криву, що дозволяє візуально оцінити продуктивність моделі.

Залежно від значення AUC, якість моделі можна умовно оцінити за наступною шкалою:

1. $\boldsymbol{AUC \geq 0.9}$ — модель із високим рівнем точності;
2. $\boldsymbol{0.8 \leq AUC < 0.9}$ —модель із високою ефективністю прогнозування;
3. $\boldsymbol{0.7 \leq AUC < 0.8}$ — модель із хорошими прогностичними характеристиками;
4. $\boldsymbol{0.6 \leq AUC < 0.7}$ —модель із прийнятним рівнем точності;
5. $\boldsymbol{0.5 \leq AUC < 0.6}$ — модель із низьким рівнем точності.

На Рис.9 зображено криву операційних характеристик нашої моделі з AUC=0,88, що дає можливість говорити про дуже добру її якість та ефективність.

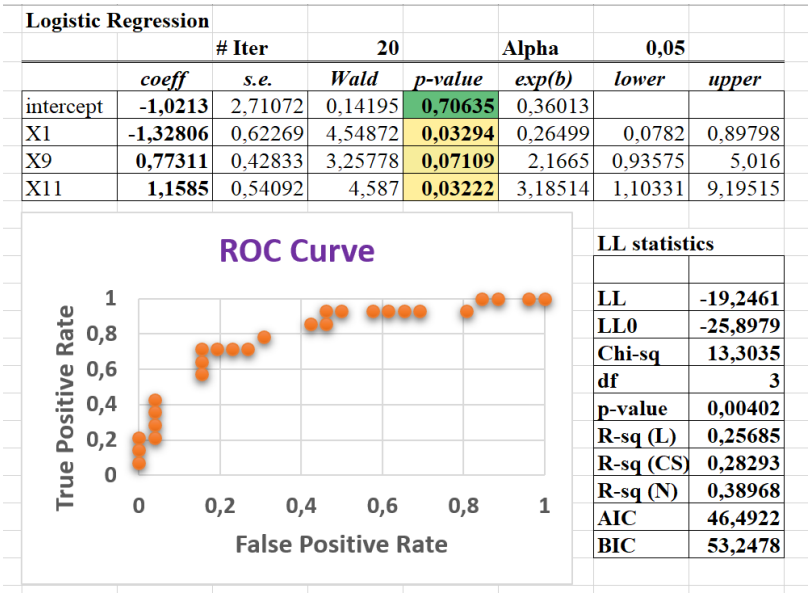


Рис. 9 Результати розрахунків для моделі логістичної регресії та ROC-крива, $Cutoff = 0{,}5$.

1.8. J-статистика Юдена

Для підвищення точності моделі візьмемо J-статистику Юдена (індекс Юдена, Youden's J statistic), це допоможе нам знайти такий поріг, при якому баланс між чутливістю і специфічністю буде максимальним. Формула для даної статистики виглядає наступним чином:

$$\text{Youden's J index} = \text{TPR} + 1 - \text{FPR},$$

У програмі MS Excel це можна зробити поставивши клік на розрахунку ROC Table і знайти максимальне значення Youden's index, далі у стовпчику p-Pred вибрати відповідне значення ймовірності – воно буде давати найкраще значення *cutoff=0,462* замість стандартного 0,5.

В нашій моделі отримали ми $Youden's\ index = 1{,}56$, тобто чутливість (TPR) 71,4%, специфічність (FPR) 84,6.

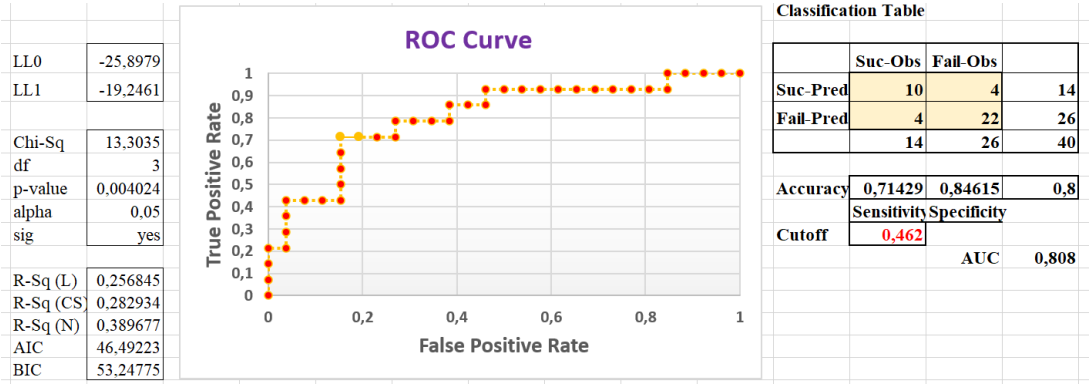


Рис.10 Результати розрахунків для моделі логістичної регресії та ROC-крива, $Cutoff = 0{,}462$.

На рис.10 ми можемо побачити новостворену модель із врахованою J-статистику. Значення критеріїв: $AIC = 46{,}5, BIC = 53{,}25$, а площа під кривою операційних характеристик моделі AUC=0,81, значення p-value не перевищує прийнятий поріг 0,5, що є свідченням того, що модель добре побудована та, за класифікацією, є моделлю із високою ефективністю прогнозування.

Розраховані коефіцієнти моделі для трьох факторів (Рис.10) дають наступну формулу прогнозування:

$$\ln\left(\frac{p}{1-p}\right) = -1{,}02 - 1{,}33X_1 + 0{,}77X_2 + 1{,}16X_3.$$

Розділ 2. Факторний аналіз

2.1. Теоретичні основи факторного аналізу

Означення 6. Факторний аналіз (*англ. factor analysis*) - це метод, який застосовують для великої кількості змінних, що зводить ці змінні до набагато меншої кількості величин, що будуть незалежними між собою, які і будуть називатися факторами.

Варто зазначити, що змінні, які відносяться до одного фактора, будуть сильно корелювати між собою, а змінні із різних факторів будуть мати слабку кореляцію. Тобто, факторний аналіз має на меті використати такі комплексні фактори, котрій максимально повно і ясно обґрунтують зв'язки між певними змінними.

Факторний аналіз має наступну процедуру, котра складається із чотирьох кроків:

1. Створення кореляційної матриці для змінних, які присутні у дослідженні;
2. Визначення групи факторів;
3. Використання методу обертання Варімакс для отримання більш чіткої структури;
4. Тлумачення факторів.

Зберемо всі змінні X_i у вектор X для кожного індивідуального суб'єкта. Нехай X_i позначає спостережувану характеристику i.

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix} - \text{вектор характеристик}$$

Це випадковий вектор із середнім значенням у генеральній сукупності. Припустимо, що вектор характеристик X вибірково отримано із генеральної сукупності з вектором середніх значень:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_p \end{pmatrix} - \text{вектор середніх значень генеральної сукупності}$$

Тоді $E(X_i) = \mu_i$ позначає середнє значення змінної i у генеральній сукупності.

Розглянемо m неспостережуваних загальних факторів $F_1, F_2, \cdots, F_m$, $m < p$

Загальні фактори також збираються у вектор:

$$\boldsymbol{F} = \begin{pmatrix} F_1 \\ \vdots \\ F_p \end{pmatrix} - \text{вектор загальних факторів}$$

Наша факторна модель може розглядатися, як серія множинних регресій, що передбачають кожну із спостережуваних змінних X_i за значенням неспостережуваних загальних факторів f_i:

$$X_1 = \mu_1 + l_{11}F_1 + L_{12}F_2 + \cdots + l_{1m}F_m + \varepsilon_1$$
$$X_2 = \mu_2 + l_{21}F_1 + L_{22}F_2 + \cdots + l_{2m}F_m + \varepsilon_2$$
$$\vdots$$
$$X_p = \mu_p + l_{p1}F_1 + L_{p2}F_2 + \cdots + l_{pm}F_m + \varepsilon_p$$

Коефіцієнти регресії l_{ij} для всіх поданих множинних регресій називаються факторним навантаженням.

В даному випадку l_{ij} – це навантаження i-ї змінної на j-й фактор. Ці значення збираються у матрицю, як показано нижче:

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & \cdots & l_{1m} \\ \vdots & \ddots & \\ l_{p1} & \cdots & l_{pm} \end{pmatrix}$$

ε_p – частина змінної X_i, яка не пояснюється факторами, або унікальний вплив. Вони також збираються у вектор:

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_p \end{pmatrix} - \text{вектор специфічних факторів}$$

Тобто, базова модель є подібною до регресійної моделі. Кожна з цих залежних змінних X передбачується як лінійна функція неспостережуваних загальних факторів $F_1, F_2, \cdots, F_m$.

Таким чином, це все зводиться до такого вигляду:

$$\boldsymbol{X = \mu + LF + \varepsilon},$$

2.2. Аналіз головних компонент

Аналіз головних компонент - це статистичний метод, який використовується для аналізу взаємозв'язків між великою кількістю змінних і для пояснення цих змінних з точки зору меншого числа змінних, званих головними компонентами, з мінімальною втратою інформації.

Якщо $X = (x_i)$ випадковий вектор. Вектор головних компонент $Y = (y_i)$ визначається фомулою:

$$y_i = \sum_{i=1}^{n} \mathrm{B}_{ij} x_i.$$

B_{ij} - матриця коефіцієнтів регресії, тоді кожен y_i є лінійною комбінацією x_i, і вектор Y є, також, випадковим вектором.

Метою аналізу є вибір значень коефіцієнтів регресії B_{ij} таким чином, щоб максимізувати дисперсію $var(y_i)$ за умови обмеження, що $cov(y_i, y_j) = 0$ для всіх $i \neq j$. За допомогою MS Excel () знаходяться

коефіцієнти B_{ij} використовуючи теорему спектрального розвинення (Spectral Decomposition Theorem):

$$\Sigma = B^{T}DB,$$

де B - є матрицею $k \times k$ стовпці якої є одиничними власними векторами $\beta_1, \beta_2, \cdots \beta_k$, що відповідають власним значенням $\lambda_1, \lambda_2, \cdots \lambda_k$ матриці Σ і D - це $k \times k$ діагональна матриця, головна діагональ якої складається з $\lambda_1, \lambda_2, \cdots \lambda_k$.

Крім того, спектральна теорема може бути виражена як:

$$\Sigma = \sum_{j=1}^{k} \lambda_j \beta_j \beta_j^{T}.$$

Властивість, яка використовується: якщо $\lambda_1 > \lambda_2 > \cdots > \lambda_k$ є власними значеннями матриці Σ з відповідними власними векторами $\beta_1, \beta_2, \cdots \beta_k$, тоді $\Sigma = \sum_{j=0}^{k} \lambda_j \beta_j \beta_j^{T}$, і крім того, для всіх i та $i \neq j$:

$$\mathrm{var}(y_i) = \lambda_i, \mathrm{cov}(y_i, y_j) = 0.$$

Оскільки вектори стовпців β_j є ортонормованими, $\beta_i \cdot \beta_j = \beta_i^T \beta_j = 0$, якщо $i \neq j$ та $\beta_i^T \beta_j = 1$, якщо $i \neq j$, таким чином:

$$\mathrm{var}(y_i) = \sum_{p=1}^{k}\sum_{m=1}^{k} \beta_{ip}\sigma_{pm}\beta_{mj} = \beta_i^T \left(\sum_{j=1}^{k} \lambda_j \beta_j \beta_j^{T} \right) \beta_i = \sum_{j=1}^{k} \lambda_j \left(\beta_i^T \beta_j\right)\left(\beta_j^T \beta_i\right)$$

$$= \lambda_j.$$

$$\mathrm{cov}(y_i, y_j) = \sum_{p=1}^{k}\sum_{m=1}^{k} \beta_{ip}\sigma_{pm}\beta_{mj} = \beta_i^T \left(\sum_{r=1}^{k} \lambda_r \beta_r \beta_r^{T} \right) \beta_i$$

$$= \sum_{j=1}^{k} \lambda_r \left(\beta_i^T \beta_r\right)\left(\beta_r^T \beta_i\right) = 0.$$

За означенням - слід матриці - це сума її власних значень. Також у коваріаційної матриці Σ головна діагональ містить значення $\sigma_1^2, \sigma_2^2, \cdots, \sigma_k^2$ тоді слід матриці trace (Σ):

$$\text{trace}(\Sigma) = \sum_{i=1}^{k} \sigma_i^2 = \sum_{i=1}^{k} \lambda_i.$$

Таким чином, загальна дисперсія $\sum_{i=1}^{k} \sigma_i^2$ для змінної X може виражатись як $\text{trace}(\Sigma) = \sum_{i=1}^{k} \lambda_i$ але це також загальна дисперсія для Y. Тепер з цього випливає, що частина повної дисперсії (X або Y), пояснена i -тим головним компонентом y_i, є $\frac{\lambda_i}{\sum_{i=1}^{k} \lambda_i}$. Припускаючи, що $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ частина загальної дисперсії пояснюється першими m головними компонентами, тому:

$$\sum \% \rightarrow \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{k} \lambda_i}.$$

Наша мета - знайти зменшену кількість головних компонентів, які можуть пояснити більшу частину загальної дисперсії, тобто ми шукаємо значення m, яке є якомога меншим, але таким, що співвідношення $\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{k} \lambda_i} \rightarrow 1$ близьке до 1.

Застосування коваріаційної матриці для вибірки

Оскільки популяційна коваріаційна матриця Σ невідома, використовуємо вибіркову коваріаційну матрицю S:

$$\text{cov } sample(X, Y) = \frac{1}{n-1}(X - \bar{X})(X - \bar{X})^T = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(x_i - \bar{x})^T}{n-1}$$

де тепер ми вважаємо $X = \left(x_{ij}\right)$ матрицею k × n такою, що для кожного i, $\left\{x_{ij} : 1 \leq j \leq n\right\}$ є випадковою вибіркою для випадкової величини x_i. Оскільки коваріаційна матриця вибірки симетрична, існує подібний

спектральний розклад, як $\Sigma = \sum_{j=1}^{k} \lambda_j \, \mathrm{B}_j \mathrm{B}_j^T$, де $\mathrm{B}_j = \left[b_{ij}\right]$ - це одиничні власні вектори S, що відповідають власним значенням λ_j з S (насправді це трохи зловживання позначеннями, оскільки ці λ_j не збігаються з власними значеннями Σ). Далі коефіцієнти регресії будуть b_{ij}, як і для популяції: $y_i = \sum_{i=1}^{n} b_{ij} x_i$ та для всіх $i \neq j$: $\mathrm{var}(y_i) = \lambda_i, \mathrm{cov}\left(y_i, y_j\right) = 0$, також: $\mathrm{trace}(S) = \sum_{i=1}^{k} s_i^2 = \sum_{i=1}^{k} \lambda_i$.

2.2.1. Кореляційна матриця

На практиці ми зазвичай віддаємо перевагу стандартизації вибіркових балів. Для цього ми використаємо кореляційну матрицю.
Нехай $R = [r_{ij}]$, де $[r_{ij}]$ – кореляція між x_i та x_j, тобто

$$r_{ij} = \frac{cov(x_i, x_j)}{s_i s_j}$$

Варто зазначити, що для підвищення точності аналізу бажано мати більшу кількість даних, це дозволить отримати більш надійні статистичні висновки та зменшити похибку оцінки кореляційних зав'язків. Кореляційна матриця буде побудована на основі відповідей, які надали 150 студентів НТТУ «КПІ ім. Ігоря Сікорського».

Вибіркова кореляційна матриця R показана на рис. 11.

Factor Analysis - Principal Component Extraction

Descriptive statistics

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14
Mean	3,87582	2,56863	1,71895	2,0719	3,81046	4,37908	3,18301	2,68627	2,90196	2,7451	2,33333	2,81699	3,28105	3,19608
Std dev	1,06574	1,46792	1,21101	1,37221	1,16275	0,94595	1,36901	1,42566	1,40846	0,96334	0,9934	1,20546	1,09098	1,08251
Skewness	-0,7731	0,41336	1,59274	0,98351	-0,9461	-1,4886	-0,2113	0,26569	0,14751	-0,1387	0,30826	0,10733	-0,4565	-0,0837
Kurtosis	-0,1038	-1,2214	1,31025	-0,4186	0,16488	1,35177	-1,1816	-1,2759	-1,2691	-0,4504	-0,4131	-0,8105	-0,3244	-0,6121

Correlation Matrix

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	
X1	1	0,33981	0,05944	0,08262	0,61797	0,47771	0,38093	0,52843	0,28549	-0,172	-0,1781	-0,0639	-0,0433	0,04406	
X2	0,33981	1	0,31624	0,33884	0,36036	0,19908	0,2949	0,38445	0,32307	-0,2458	-0,3429	-0,3349	-0,2483	-0,2859	
X3	0,05944	0,31624	1	0,25374	0,19086	0,11659	0,02726	0,15818	0,24603	-0,1746	-0,1951	-0,2247	-0,1988	-0,1384	
X4	0,08262	0,33884	0,25374	1	0,09931	0,10051	0,26261	0,13603	0,21132	-0,0856	-0,1721	-0,1034	-0,0971	-0,0937	
X5	0,61797	0,36036	0,19086	0,09931	1	0,39473	0,30711	0,42427	0,31799	-0,3841	-0,3607	-0,1798	-0,0511	-0,1271	
X6	0,47771	0,19908	0,11659	0,10051	0,39473	1	0,33725	0,31317	0,20091	-0,0738	-0,1003	-0,0715	-0,0848	-0,0474	
X7	0,38093	0,2949	0,02726	0,26261	0,30711	0,33725	1	0,53186	0,52116	-0,1689	-0,4031	-0,143	0,00058	-0,1087	
X8	0,52843	0,38445	0,15818	0,13603	0,42427	0,31317	0,53186	1	0,3843	-0,279	-0,3019	-0,137	-0,091	-0,0323	
X9	0,28549	0,32307	0,24603	0,21132	0,31799	0,20091	0,52116	0,3843	1	-0,3192	-0,4655	-0,278	-0,1875	-0,1685	
X10	-0,172	-0,2458	-0,1746	-0,0856	-0,3841	-0,0738	-0,1689	-0,279	-0,3192	1	0,5156	0,33348	0,18129	0,19335	
X11	-0,1781	-0,3429	-0,1951	-0,1721	-0,3607	-0,1003	-0,4031	-0,3019	-0,4655	0,5156	1	0,51276	0,17402	0,30589	
X12	-0,0639	-0,3349	-0,2247	-0,1034	-0,1798	-0,0715	-0,143	-0,137	-0,278	0,33348	0,51276	1	0,59965	0,5873	
X13	-0,0433	-0,2483	-0,1988	-0,0971	-0,0511	-0,0848	0,00058	-0,091	-0,1875	0,18129	0,17402	0,59965	1	0,52124	
X14	0,04406	-0,2859	-0,1384	-0,0937	-0,1271	-0,0474	-0,1087	-0,0323	-0,1685	0,19335	0,30589	0,5873	0,52124	1	
	1,311	1,27247	0,48192	0,40396	1,41801	0,72958	1,28765	1,3763	1,31124	0,93627	1,48018	1,41002	0,86012	0,91885	15,1976

Рис. 11 Описова статистика вибірки та кореляційна матриця

Всі значення на головні діагоналі дорівнюють 1, як ми і очікували, оскільки всі дисперсії були стандартизовані.

Аналогічно до популяції, припускаючи, що $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$, MS Excel Factor Analysis знаходить значення m, так що $\sum_{i=1}^{m} \lambda_i$ і пояснює якомога більшу частину загальної дисперсії. Таким чином, ми зменшуємо кількість основних компонентів, необхідних для пояснення більшості мінливості.

Загальна дисперсія для 14 випадкових величин дорівнює 14 (оскільки дисперсія була стандартизована до 1 в кореляційній матриці), що, як і очікувалося, дорівнює сумі 14 власних значень, що і виводиться MS Excel Factor Analysis для $m = 4, \sum_{i=1}^{m} \lambda_i = 62\%$, $m = 5, \sum_{i=1}^{m} \lambda_i = 69\%$

2.2.2. Часткова кореляція та тест Kaiser-Meyer-Olkin

При великій кількості змінних та великій вибірці вірогідна така ситуація, що певні змінні не дуже добре корелюють з іншими змінними, тому для цієї мети ми будемо застосовувати часткову кореляційну матрицю та тест Кайзера-Майєра-Олкіна (**Kaiser-Meyer-Olkin, KMO).**

Щоб обчислити часткову кореляційну матрицю, необхідно знайти обернену кореляційну матрицю, як зображено на рис. 12.

Inverse of Correlation Matrix

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	sqrt(diag)
X1	2,2373	-0,2747	0,2198	0,0658	-1,0061	-0,4507	-0,1317	-0,5345	-0,0766	-0,1394	-0,1603	0,0281	0,1453	-0,3505	1,4958
X2	-0,2747	1,5996	-0,2368	-0,3492	-0,1297	0,0535	0,0077	-0,2761	-0,0412	-0,0174	0,1053	0,149	0,0413	0,2371	1,2648
X3	0,2198	-0,2368	1,2721	-0,2303	-0,1608	-0,1224	0,3227	-0,1391	-0,2473	0,0016	0,0251	0,0976	0,0736	-0,0639	1,1279
X4	0,0658	-0,3492	-0,2303	1,2353	0,0501	-0,0073	-0,3186	0,1235	-0,0127	-0,0028	0,0299	-0,0989	0,0739	-0,0347	1,1114
X5	-1,0061	-0,1297	-0,1608	0,0501	2,0668	-0,2625	0,1251	-0,0691	-0,0523	0,4122	0,2644	-0,0611	-0,2489	0,2475	1,4376
X6	-0,4507	0,0535	-0,1224	-0,0073	-0,2625	1,4125	-0,3317	0,0145	0,0483	-0,0745	-0,1318	-0,007	0,1144	0,0228	1,1885
X7	-0,1317	0,0077	0,3227	-0,3186	0,1251	-0,3317	2,0623	-0,6884	-0,6585	-0,2207	0,4425	-0,0445	-0,2946	0,201	1,4361
X8	-0,5345	-0,2761	-0,1391	0,1235	-0,0691	0,0145	-0,6884	1,8579	-0,0391	0,2241	-0,0078	-0,069	0,1414	-0,1666	1,3631
X9	-0,0766	-0,0412	-0,2473	-0,0127	-0,0523	0,0483	-0,6585	-0,0391	1,6891	0,1392	0,3388	-0,0211	0,2138	-0,0659	1,2997
X10	-0,1394	-0,0174	0,0016	-0,0028	0,4122	-0,0745	-0,2207	0,2241	0,1392	1,5366	-0,5912	-0,0831	-0,095	0,0388	1,2396
X11	-0,1603	0,1053	0,0251	0,0299	0,2644	-0,1318	0,4425	-0,0078	0,3388	-0,5912	2,1296	-0,8591	0,3695	-0,0494	1,4593
X12	0,0281	0,149	0,0976	-0,0989	-0,0611	-0,007	-0,0445	-0,069	-0,0211	-0,0831	-0,8591	2,4275	-0,9244	-0,6381	1,5581
X13	0,1453	0,0413	0,0736	0,0739	-0,2489	0,1144	-0,2946	0,1414	0,2138	-0,095	0,3695	-0,9244	1,8667	-0,5198	1,3663
X14	-0,3505	0,2371	-0,0639	-0,0347	0,2475	0,0228	0,201	-0,1666	-0,0659	0,0388	-0,0494	-0,6381	-0,5198	1,7624	1,3276

Рис. 12 Обернена кореляційна матриця

Часткова кореляція між змінними між x_i та x_j, де $i \neq j$,зберігаючи всі інші змінні незміненими, визначається за формулою

$$r_{x_i x_j, z} = -\frac{p_{ij}}{\sqrt{p_{ii}\, p_{jj}}}$$

де z це список змінних $x_1, \dots, x_k$ за винятком x_i та x_j, а обернена кореляційна матриця дорівнює $R^{-1} = [p_{ij}]$.

Побудована часткова кореляційна матриця, показана на рис.13.

Також ми будемо застосовувати тест Кайзера-Майєра-Олкіна (KMO) міра адекватності вибірки (MSA) для змінної x_i визначається за формулою

$$KMO_j = \frac{\Sigma_{i \neq j} r_{ij}^2}{\Sigma_{i \neq j} r_{ij}^2 + \Sigma_{i \neq j} u_{ij}^2}$$

де обернена кореляційна матриця $R = [p_{ij}]$, а часткова кореляційна матриця $U = [u_{ij}]$. Загальна міра КМО адекватності вибірки визначається наведеною вище формулою для всіх комбінацій $i \neq j$

КМО приймає значення від 0 до 1. Значення біля 0 вказує на те, що сума часткових кореляцій велика порівняно з сумою кореляцій. Це вказує на те, що кореляції широко поширені і тому не групуються серед кількох змінних, що вказувало б на проблему для факторного аналізу. Значення близько 1 вказує на хорошу придатність для факторного аналізу.

Результати тесту КМО зображені на рис.13

Partial Correlation Matrix

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	
X1	1	0,145192	-0,130307	-0,039564	0,467863	0,253514	0,061316	0,262154	0,039406	0,075169	0,073452	-0,012068	-0,071117	0,176524	
X2	0,145192	1	0,166008	0,248452	0,07136	-0,035569	-0,004264	0,16015	0,025039	0,011097	-0,057028	-0,075594	-0,023927	-0,141221	
X3	-0,130307	0,166008	1	0,183748	0,099153	0,091282	-0,199249	0,09048	0,168683	-0,001142	-0,015223	-0,055532	-0,047746	0,042691	
X4	-0,039564	0,248452	0,183748	1	-0,031372	0,005543	0,199603	-0,081552	0,00878	0,002063	-0,018437	0,0571	-0,048689	0,023512	
X5	0,467863	0,07136	0,099153	-0,031372	1	0,153641	-0,060571	0,035278	0,027982	-0,231307	-0,126003	0,027267	0,126699	-0,129667	
X6	0,253514	-0,035569	0,091282	0,005543	0,153641	1	0,194371	-0,008932	-0,03126	0,05054	0,076001	0,003798	-0,070437	-0,014476	
X7	0,061316	-0,004264	-0,199249	0,199603	-0,060571	0,194371	1	0,351682	0,352839	0,123985	-0,211153	0,019903	0,150168	-0,105429	
X8	0,262154	0,16015	0,09048	-0,081552	0,035278	-0,008932	0,351682	1	0,022072	-0,132661	0,003912	0,032481	-0,075901	0,092077	
X9	0,039406	0,025039	0,168683	0,00878	0,027982	-0,03126	0,352839	0,022072	1	-0,086411	-0,178649	0,010431	-0,120406	0,038212	
X10	0,075169	0,011097	-0,001142	0,002063	-0,231307	0,05054	0,123985	-0,132661	-0,086411	1	0,326834	0,043009	0,05612	-0,02357	
X11	0,073452	-0,057028	-0,015223	-0,018437	-0,126003	0,076001	-0,211153	0,003912	-0,178649	0,326834	1	0,377843	-0,185302	0,025506	
X12	-0,012068	-0,075594	-0,055532	0,0571	0,027267	0,003798	0,019903	0,032481	0,010431	0,043009	0,377843	1	0,434252	0,308523	
X13	-0,071117	-0,023927	-0,047746	-0,048689	0,126699	-0,070437	0,150168	-0,075901	-0,120406	0,05612	-0,185302	0,434252	1	0,286604	
X14	0,176524	-0,141221	0,042691	0,023512	-0,129667	-0,014476	-0,105429	0,092077	0,038212	-0,02357	0,025506	0,308523	0,286604	1	
	0,444238	0,172624	0,180225	0,151169	0,367095	0,149856	0,466963	0,26761	0,212902	0,21465	0,391961	0,442899	0,382307	0,27009	4,114589

КМО

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	
0,746907	0,880545	0,727819	0,727686	0,794356	0,829599	0,733865	0,837211	0,860314	0,813498	0,790634	0,760972	0,69229	0,772831	0,786943

Рис.13 Часткова кореляційна матриця та тест КМО

Власні значення та власні вектори (Eigenvalues and eigenvectors)

Для подальшого дослідження ми застосуємо такі поняття, як власні значення та власні вектори.

Означення 7. Власне значення (eigenvalues) квадратної $k \times k$ матриці A є скалярною величиною λ і такою, що $\det(A - \lambda I) = 0$, де I являє собою одиничну матрицю $k \times k$. Ненульовий вектор-стовпець

X є власним вектором **(eigenvectors)**, який відповідає власному значенню λ за умови $AX = \lambda X$.

На рис. 14 ми можемо побачити зображення власних значень та власних векторів

Eigenvalues and eigenvectors

4,333407	2,134613	1,174468	1,137745	0,975964	0,73074	0,68054	0,573931	0,513003	0,458198	0,449605	0,294424	0,280237	0,263124
-0,275235	-0,373318	-0,216045	-0,266418	-0,071392	-0,131827	-0,12268	-0,064487	-0,298862	-0,039594	-0,271599	-0,218495	-0,586514	-0,262965
-0,315148	0,037879	0,252747	-0,185031	-0,089346	-0,37418	-0,387626	0,333214	0,046215	0,558194	0,250902	-0,075057	0,095588	0,020303
-0,185366	0,128168	0,445529	-0,179686	-0,521635	0,540557	-0,041577	0,159484	0,165102	-0,212905	-0,069747	-0,198622	-0,080676	-0,059584
-0,168556	0,014209	0,68977	-0,064622	0,113958	-0,420412	0,329955	-0,331913	-0,101466	-0,174108	-0,121303	0,152185	-0,057151	0,030143
-0,314779	-0,226272	-0,231865	-0,089156	-0,349502	-0,196069	0,14404	0,195386	-0,329324	-0,352247	-0,010906	0,043147	0,531851	0,226073
-0,214774	-0,270718	-0,141755	-0,441913	0,092107	0,224597	0,580795	-0,028228	0,322535	0,331504	0,140006	0,184614	0,020943	0,010478
-0,295621	-0,236754	0,101118	0,206992	0,522026	0,097056	0,028989	0,096968	0,182751	-0,167051	0,039834	-0,645143	0,1577	0,085678
-0,310165	-0,267483	-0,032239	0,014252	0,089827	0,060819	-0,523126	-0,267546	0,430216	-0,260344	-0,018975	0,464353	0,044274	0,056483
-0,318623	-0,040193	0,130345	0,281992	0,197258	0,461338	-0,022948	-0,008098	-0,591106	0,174653	0,254114	0,289265	-0,102432	0,09679
0,264476	-0,096324	0,191659	-0,412682	0,422091	0,145815	-0,152519	0,456636	-0,167821	-0,083577	-0,404116	0,220793	0,176546	-0,064471
0,327603	-0,132587	0,063909	-0,454266	0,031853	0,047748	-0,192636	-0,248555	-0,150494	-0,142011	0,436459	-0,187426	-0,11694	0,529487
0,275634	-0,432829	0,175295	0,042114	-0,079632	-0,004585	-0,022305	-0,032956	-0,066845	-0,102087	0,454523	0,001121	0,206277	-0,657423
0,192564	-0,4258	0,13573	0,352765	-0,134985	-0,128221	0,149206	0,499541	0,178855	-0,07336	0,064959	0,175273	-0,396015	0,32116
0,203717	-0,439731	0,151943	0,172156	-0,219015	0,13719	-0,088708	-0,329033	-0,057873	0,456172	-0,439141	-0,134029	0,268779	0,18708

Рис.14 Власні значення та вектори

Далі застосуємо формулу $b_{ij} = \sqrt{\lambda_j} c_{ij}$, де $c_1, \dots, c_k$ є власними векторами, що відповідають власним значенням $\lambda_1 \geq \cdots \geq \lambda_k$, ми обчислимо коефіцієнти факторного навантаження (див. рис 15), що відповідають 14 факторам.

Full Load Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
X1	-0,572952	-0,545429	-0,234134	-0,284175	-0,070529	-0,11269	-0,101205	-0,048854	-0,214058	-0,026802	-0,182114	-0,118557	-0,310486	-0,134889
X2	-0,656038	0,055342	0,273909	-0,197363	-0,088266	-0,319862	-0,319771	0,252437	0,033101	0,377843	0,168236	-0,040726	0,050602	0,010415
X3	-0,385874	0,187258	0,482833	-0,191662	-0,515327	0,462086	-0,034299	0,120822	0,118253	-0,144116	-0,046767	-0,107774	-0,042708	-0,030564
X4	-0,350881	0,02076	0,747524	-0,068929	0,11258	-0,359382	0,272196	-0,251452	-0,072674	-0,117854	-0,081337	0,082577	-0,030254	0,015462
X5	-0,655271	-0,33059	-0,251279	-0,095099	-0,345276	-0,167606	0,118826	0,148021	-0,235876	-0,238437	-0,007313	0,023412	0,281548	0,115966
X6	-0,447092	-0,395528	-0,153624	-0,471367	0,090994	0,191993	0,479126	-0,021385	0,231013	0,224396	0,093878	0,100173	0,011087	0,005375
X7	-0,615389	-0,345906	0,109584	0,220789	0,515714	0,082967	0,023914	0,073461	0,130894	-0,113077	0,02671	-0,35006	0,083482	0,043949
X8	-0,645666	-0,390801	-0,034939	0,015202	0,088741	0,05199	-0,431552	-0,202688	0,308139	-0,176227	-0,012723	0,251962	0,023437	0,028973
X9	-0,663273	-0,058723	0,141258	0,300787	0,194873	0,394367	-0,018931	-0,006135	-0,423375	0,118223	0,17039	0,156958	-0,054225	0,049649
X10	0,550556	-0,140732	0,207706	-0,440188	0,416987	0,124647	-0,12582	0,345939	-0,120201	-0,056574	-0,27097	0,119804	0,093459	-0,033071
X11	0,681965	-0,193714	0,06926	-0,484544	0,031468	0,040817	-0,158915	-0,188301	-0,10779	-0,096128	0,292657	-0,101699	-0,061905	0,271604
X12	0,573782	-0,632377	0,189972	0,044921	-0,078669	-0,003919	-0,0184	-0,024967	-0,047877	-0,069103	0,304769	0,000608	0,109198	-0,337229
X13	0,400858	-0,622108	0,147094	0,376277	-0,133353	-0,109607	0,123087	0,378443	0,128104	-0,049658	0,043557	0,095105	-0,20964	0,164741
X14	0,424074	-0,64246	0,164665	0,18363	-0,216367	0,117275	-0,07318	-0,24927	-0,041451	0,308784	-0,294455	-0,072725	0,142285	0,095964

Рис.15 Факторні навантаження

2.2.3. Діаграма Scree Plot

Як зазначалося раніше, однією з головних цілей факторного аналізу є зменшення кількості параметрів. Кількість параметрів у вихідній моделі дорівнює кількості унікальних елементів у коваріаційній матриці. Враховуючи симетрію, існує $C(k, 2) = k(k + 1)/2$ таких елементів. Модель факторного аналізу вимагає $k(m + 1)$ елементів; тобто кількість параметрів у L(саме km) разом із кількістю елементів у $X = \mu + LY + \varepsilon$ (саме k).

Таким чином ми бажаємо мати таке значення для m, щоб $k(m + 1) \leq k(k + 1)/2$, тобто $m \leq (k - 1)/2$. Для нашого завдання це буде виглядати таким чином $m \leq \frac{k-1}{2} = m \leq \frac{14-1}{2} = 6$. Ми вважаємо за краще використовувати менше 6 множників, якщо це можливо.

Загалом, фактори, які мають високе значення, слід зберегти, а ті, що мають низьке власне значення, слід усунути. Загальний підхід (Кайзер) полягає у збереженні факторів із власним значенням ≥ 1 та виключенні факторів із власним значенням < 1. Це може бути прийнятним для менших моделей, але це може бути надто обмежувальним для моделей із великою кількістю змінних.

Інший підхід полягає у створенні діаграми Scree plot, тобто графіка власних значень (вісь y), усіх факторів (вісь x), де фактори перераховані в порядку зменшення їхніх власних значень (як ми робили в аналізі головних компонент).

Scree Plot

eValue	%	Cum %
4,333407	30,95%	30,95%
2,134613	15,25%	46,20%
1,174468	8,39%	54,59%
1,137745	8,13%	62,72%
0,975964	6,97%	69,69%
0,73074	5,22%	74,91%
0,68054	4,86%	79,77%
0,573931	4,10%	83,87%
0,513003	3,66%	87,53%
0,458198	3,27%	90,80%
0,449605	3,21%	94,02%
0,294424	2,10%	96,12%
0,280237	2,00%	98,12%
0,263124	1,88%	100,00%
14		

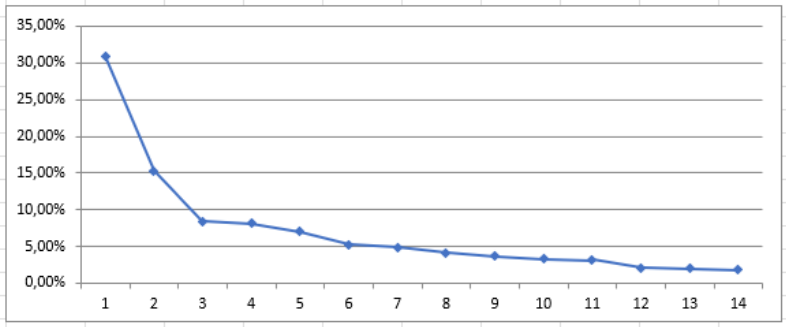


Рис. 16 Графік Scree Plot

Значення в стовпці eValue — це власні значення, перелічені в першому стовпці на рис.16 (власні значення та вектори). Кожна клітинка в стовпці % містить відсоток дисперсії, врахований відповідним власним значенням, тому ми бачимо, що 30,95% загальної дисперсії припадає на найбільше власне значення. Стовпець Cum% містить кумулятивні ваги, тому ми бачимо, що перші чотири власні значення складають 62,72% дисперсії.

Використовуючи можливість побудови діаграм Excel, можемо нанести значення в стовпці % на рис. (там де сам графік), щоб отримати графічне представлення, яке називається Scree plot.

Сукупність усіх даних прийомів полягає у збереженні всіх факторів вище (тобто ліворуч) від точки перегину (тобто точки, де крива починає вирівнюватися) та усуненні будь-якого фактора нижче (тобто праворуч) точки перегину. Оскільки крива не обов'язково гладка, може бути кілька точок перегину, тому фактична точка зрізу може бути суб'єктивною. На самому графіку є 2 точки перегину: одна при значенні 3, а інша при значенні 6.

Для наших цілей ми вирішили залишити фактори, що відповідають власним значенням, зліва від власного значення 6, тобто 5 найбільших власних значень.

2.2.4. Факторні навантаження

Факторні навантаження зображені на рис.17, який містить коефіцієнти, які обмежені чотирма найвищими загальними значеннями.

Factor Matrix (unrotated)

	1	2	3	4	5	Commun	Specific
X1	-0,572952	-0,545429	-0,234134	-0,284175	-0,070529	0,766315	0,233685
X2	-0,656038	0,055342	0,273909	-0,197363	-0,088266	0,555218	0,444782
X3	-0,385874	0,187258	0,482833	-0,191662	-0,515327	0,719389	0,280611
X4	-0,350881	0,02076	0,747524	-0,068929	0,11258	0,699766	0,300234
X5	-0,655271	-0,33059	-0,251279	-0,095099	-0,345276	0,73007	0,26993
X6	-0,447092	-0,395528	-0,153624	-0,471367	0,090994	0,6104	0,3896
X7	-0,615389	-0,345906	0,109584	0,220789	0,515714	0,825072	0,174928
X8	-0,645666	-0,390801	-0,034939	0,015202	0,088741	0,578936	0,421064
X9	-0,663273	-0,058723	0,141258	0,300787	0,194873	0,591781	0,408219
X10	0,550556	-0,140732	0,207706	-0,440188	0,416987	0,733703	0,266297
X11	0,681965	-0,193714	0,06926	-0,484544	0,031468	0,743172	0,256828
X12	0,573782	-0,632377	0,189972	0,044921	-0,078669	0,773423	0,226577
X13	0,400858	-0,622108	0,147094	0,376277	-0,133353	0,728709	0,271291
X14	0,424074	-0,64246	0,164665	0,18363	-0,216367	0,700243	0,299757
	4,333407	2,134613	1,174468	1,137745	0,975964	9,756197	4,243803

Рис. 17 Факторні навантаження

На наведених результатах видно, що, наприклад, змінна X10, X11, X12 тісно корелюють із першим фактором. В ідеалі ми хотіли б бачити, щоб кожна змінна сильно корелювала лише з одним фактором. Як ми можемо бачити на рис.17, що змінна X10 корелює як з Фактором 1, так і з Фактором 5. Ми спробуємо прояснити аналіз за допомогою методу обертання Varimax.

Факторні навантаження після застосування методу Varimax.

Нехай U – ортогональна матриця розміром $m \times m$, отже за визначенням $U^T U = I$. $L' = LU^T$ та $Y' = UY$. Тоді L' є матрицею $(k \times m) \times (m \times m) = k \times m$, а Y' являє собою $(m \times m) \times (m \times 1) = m \times 1$ вектор стовпець.

$$X = \mu + LY + \varepsilon = \mu + LU^T UY + \varepsilon = \mu + L'Y' + \varepsilon$$

$$E[Y'] = E[UY] = UE[Y] = Uo = o$$

$$var(Y') = var(UY) = Uvar(Y)U^T = UIU^T = UU^T = I$$

$$cov(Y', \varepsilon) = cov(UY, \varepsilon) = Ucov(Y, \varepsilon) = Uo = o$$

Це показує, що якщо L та Y задовільняють модель, то L' та Y' також. Оскільки існує нескінченна кількість ортогональних матриць U, існує нескінченна кількість альтернативних моделей.

Обертання початкових осей визначається ортогональною матрицею U з $det = 1$. Таким чином, заміна Y і на Y' еквівалентна обертанню осей. Це не змінить загальну дисперсію, пояснену моделлю, але це змінить розподіл дисперсій між факторами.

Ми шукаємо матрицю обертання розміром $m \times m$ та $U = [u_{ij}$ так, щоб рядки представляли існуючі фактори, а стовпці — нові фактори. Найпопулярніший підхід обертання називається **Varimax**, який максимізує різницю між факторами навантаження, зберігаючи ортогональні осі. Varimax намагається максимізувати значення V, де

$$V = \frac{1}{k}\sum_{j=1}^{m}[\sum_{i=1}^{k}(\frac{{b_{ij}}^2}{\varphi_i})^2 - (\frac{1}{k}\sum_{i=1}^{k}\frac{{b_{ij}}^2}{\varphi_i})^2]$$

Існують також неортогональні повороти, які краще справляються з диференціацією факторів, але ціною втрати ортогональності.

Factor Matrix (rotated Varimax)

	1	2	3	4	5	Commun	Specific
X1	-0,85123	-0,04479	0,0261	0,1004	0,17017	0,76631	0,23369
X2	-0,33785	0,28578	0,53023	0,14972	0,2363	0,55522	0,44478
X3	-0,09979	0,11013	0,78607	0,2323	-0,1595	0,71939	0,28061
X4	0,05475	0,02169	0,71041	-0,1472	0,41224	0,69977	0,30023
X5	-0,71774	0,02325	0,12243	0,44387	0,04865	0,73007	0,26993
X6	-0,74372	0,1024	0,05247	-0,16489	0,12981	0,6104	0,3896
X7	-0,31392	0,0192	0,01005	0,05098	0,85056	0,82507	0,17493
X8	-0,56311	-0,00697	0,10303	0,2029	0,45827	0,57894	0,42106
X9	-0,16478	0,1408	0,18668	0,3323	0,63208	0,59178	0,40822
X10	0,11571	-0,13117	-0,07972	-0,83094	-0,07937	0,7337	0,2663
X11	0,03681	-0,2751	-0,08995	-0,67418	-0,45114	0,74317	0,25683
X12	0,01852	-0,80384	-0,09023	-0,32149	-0,12421	0,77342	0,22658
X13	0,07707	-0,83959	-0,12473	0,00927	0,04706	0,72871	0,27129
X14	-0,02396	-0,82581	-0,04399	-0,08502	-0,09244	0,70024	0,29976
	2,38436	2,25346	1,50781	1,74216	1,86841	9,7562	4,2438

Рис. 18 Факторні навантаження після обертання Varimax

Нові факторні навантаження, які зображені на рис.18, дають змогу зробити висновки, щодо приналежності факторів кожного із питань в анкеті.

Наведемо список питань:

X1 На Вашу думку, чи стали для Вас події, пов'язані із російським вторгненням в Україну, значним стресовим досвідом?

X2 Чи змінились Ваші плани на навчання в Україні у зв'язку з російським вторгненням?

X3 Чи були думки відкласти вступ до ЗВО на наступний рік?

X4 Чи змінило російське вторгнення в Україну Ваші плани на вибір майбутньої професії?

X5 Наскільки стресовим став навчальний процес у зв'язку з російським вторгненням в Україну в школі/в університеті?

X6 Чи відчували Ви сильні негативні емоції (страх або злість), пов'язані із початком російського вторгненням в Україну?

X7 Чи відчували Ви під час російського вторгнення проблеми у переживанні позитивних емоцій (наприклад, неможливість відчувати радість або любов)?

X8 Чи були у Вас сильні фізичні реакції (серцебиття, утруднене дихання, потіння), коли щось нагадує про події російського вторгнення?

X9 Чи відчули Ви втрату інтересу до тієї активності (діяльності), яка раніше приносила задоволення?

X10 Оцініть свою фізичну спроможність щодо навчання під час російської агресії?

X11 Оцініть свій моральний настрій щодо навчання під час російської агресії?

X12 Оцініть рівень своєї мотивації до навчання в Україні в наступні 2-3-4 роки?

X13 Оцініть свої подальші перспективи, щодо фахової роботи в Україні після закінчення війни з московитами перемогою України?

X14 Чи бачите Ви перспективу в навчанні в Україні на найближчі (1-2-3) роки?

На рис. 19 відображено, які саме питання відносяться до кожного із факторів.

Емоційний стрес	Мотивація, бачення майбутнього	Вибір майбутньої професії	Стрес у навчанні	Втрата позитивних емоцій
Фактор 1	Фактор 2	Фактор 3	Фактор 4	Фактор 5
X1	X12	X2	X5	X4
X5	X13	X3	X11	X7
X6	X14	X4		X8
X8				X9

Рис. 19 Розподіл питань за факторами відповідно до результатів факторного аналізу.

Результати даного факторного аналізу підтверджені дослідженням, що було реалізовано за допомогою інструментів аналізу даних мови програмування Python (детальне виконання відображене у Додатку 1).

Розділ 3. Застосування факторного аналізу для даних опитування

3.1. Опис завдання та даних:

У 2024 році було проведено масштабне опитування людей щодо стану здоров’я, звичок, місця проживання, наявності захворювань тощо. В даному дослідженні прийняли участь 1101 пацієнт, кожному з яких було запропоновано відповісти на 165 запитань.

Основною метою даного опитування було виявити зв'язок між вживанням вершкового масла та рослинної олії й загальним станом здоров’я людини. Складність завдання полягала в тому, що була пропущена велика кількість відповідей на окремі запитання та текстовий формат даних, що унеможливлювало використання інструментів аналізу даних мовою програмування Python.

Для аналізу та обробки даних буде використано мову програмування Python у середовищі Jupyter Notebook. На першому етапі, спробуємо розглянути дані у початковому вигляді, без форматування та стандартизації (детальний програмний код проведеного аналізу міститься в Додатку 2).

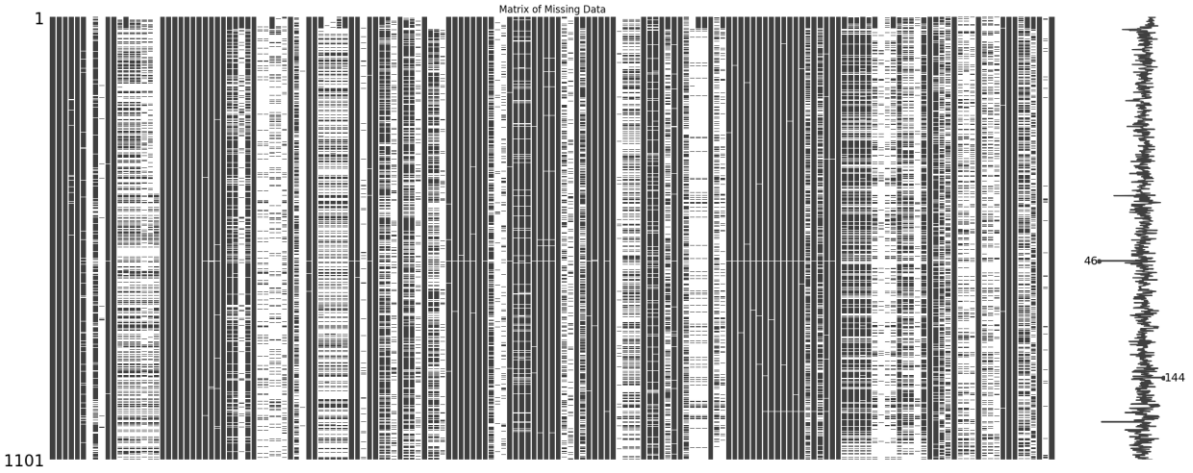


Рис. 20 Діаграма пропущених значень

На рисунку 20 зображені дані у вигляді стовпців, довжина кожного стовпця сягає від 1 до 1101, що відповідає кількості респондентів. Чорним кольором позначені комірки із відсутніми даними. Виявлення значної кількості пропущених значень підкреслює необхідність попередньої обробки даних перед проведенням аналізу.

3.2. Форматування даних

В даному розділі буде опрацьовано дані за допомогою мови програмування Python, оскільки ця мова програмування передбачає достатньо швидке та точне обчислення, що є важливим для роботи із великими обсягами даних.

Перед початком дослідження та виконанням розрахунків, були поставлені наступні завдання:

1. Видалити ті стовпці, які не мають значень або в яких майже не містяться даних;
2. Перетворити категоріальні дані в числові,

3. Замінити дані в стовпцях із текстом та категоріальними даними на числа, оскільки для проведення кореляцій та факторного аналізу необхідні саме числові змінні;
4. Заповнити пропуски в стовпцях із текстом найбільш частим значенням;
5. Закодувати текстові значення за допомогою LabelEncoder;
6. Заповнити пропущені значення в стовпцях із числами середнім значенням;
7. Масштабувати дані.

За допомогою бібліотек *pandas, sklearn.impute, sklearn.preprocessing, sklearn.decomposition, matplotlib.pyplot, seaborn* виконуємо поставлені завдання

.

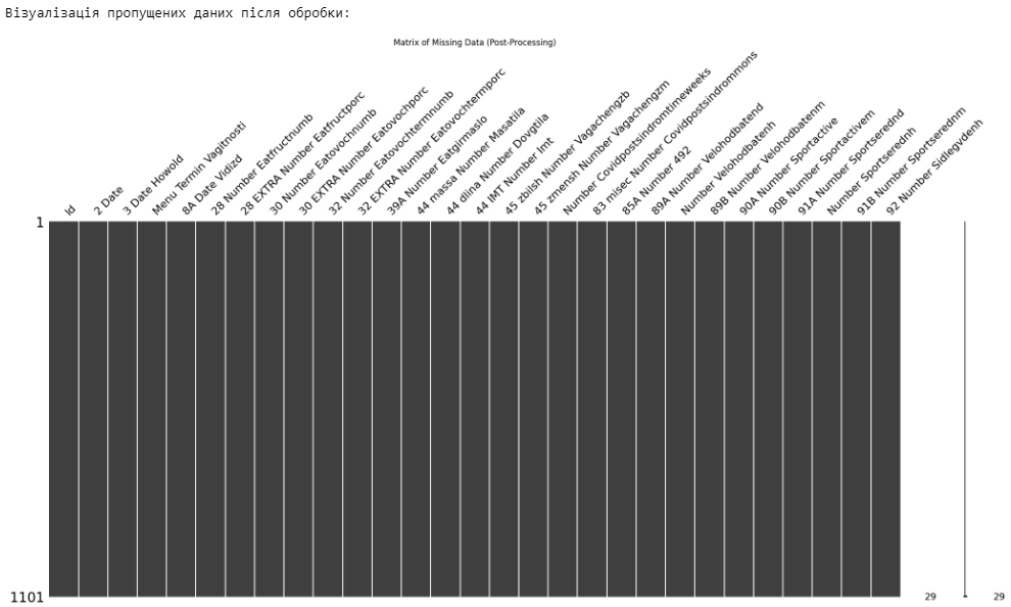


Рис.21 Діаграма, яка відображає наповненість стовпців даними

Рис. 21 демонструє, що пусті місця в даних були успішно заповнені, а також була зменшена кількість стовпців, оскільки деякі із них не мали впливу на результат (наприклад, дата проходження тесту). Згідно із цією

діаграмою, можна зробити висновок, що дані тепер готові до подальшого аналізу.

3.3. Побудова кореляційної матриці

Перш ніж, як проводити аналіз даних та факторний аналіз, нам необхідно запевнитися, що існують кореляції між змінними у наборі даних. Для цього побудуємо кореляційну матрицю, яка дозволить виявити взаємозв'язки між різними змінними.

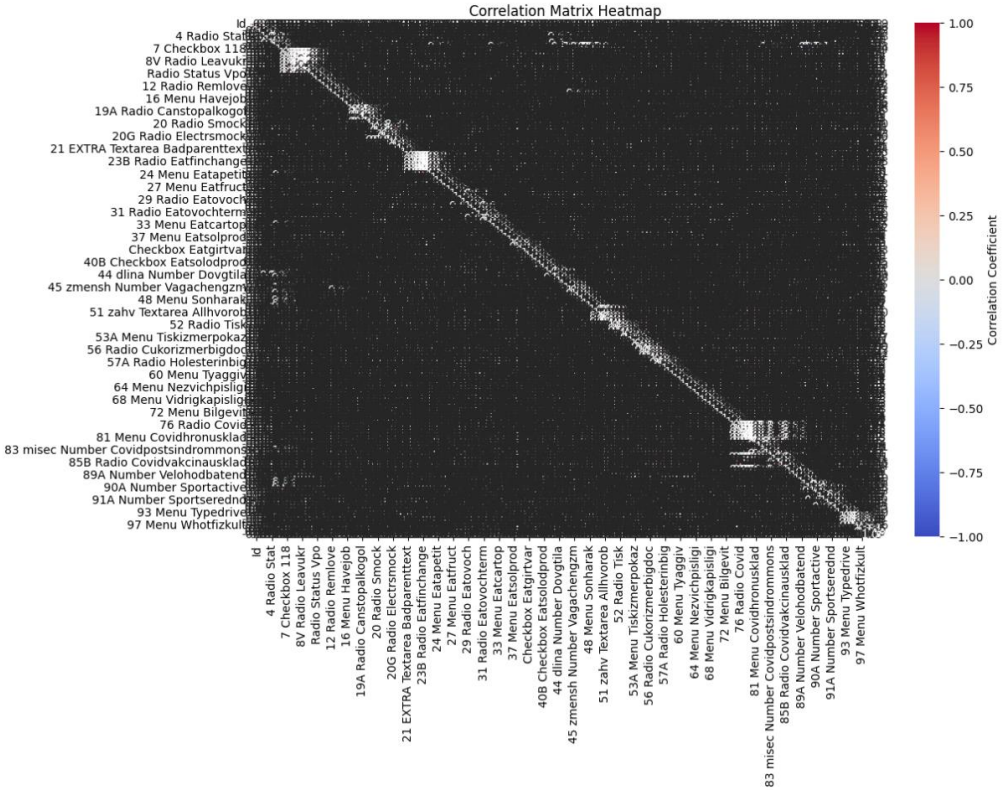


Рис. 22 Теплова карта кореляційної матриці.

На рисунку 22 відображена кореляційна матриця у вигляді теплової карти, яка показує ступінь зв’язку між змінними. Але проблема полягає в тому,

що кількість поданих кореляцій настільки велика, що ускладнює визначення чітких взаємозв’язків між даними.

Спробуємо відобразити лише ті кореляції, модуль коефіцієнтів яких більше за 0.5, для того, щоб виділити лише найбільш значущі зв’язки.

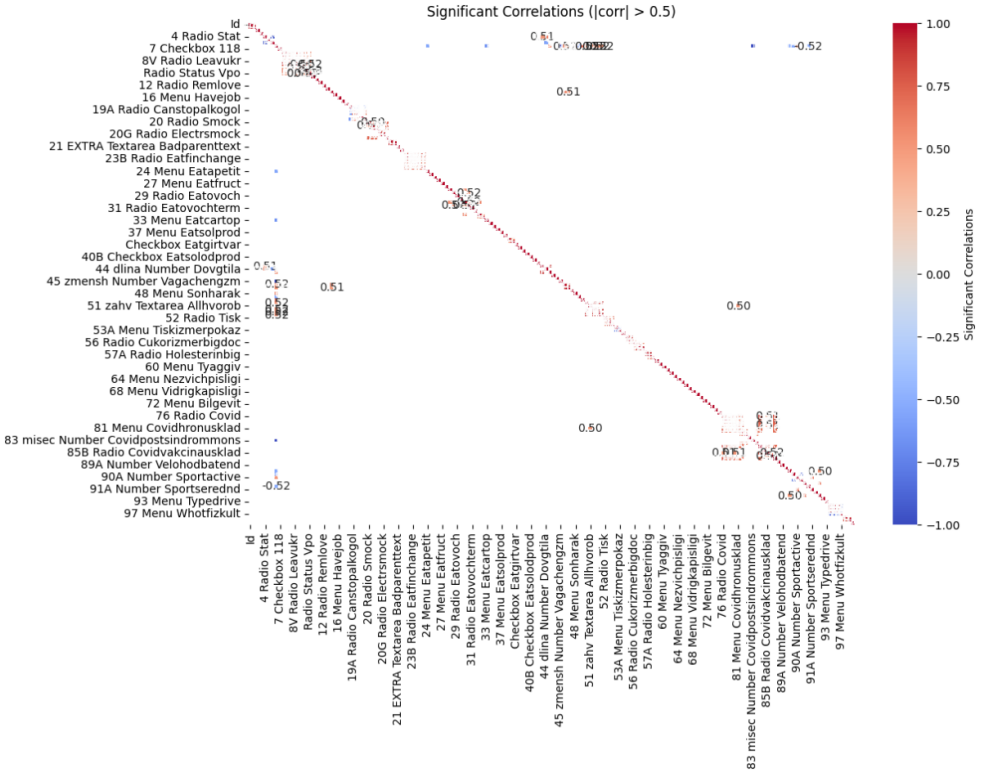


Рис. 23 Теплова карта кореляційної матриці із $|corr| > 0.5$

На представленому рисунку 23, можемо побачити, що наявні значущі кореляції із модулем коефіцієнтів, що більші за 0.5. Отже, робимо висновок, що кореляції дійсно існують, тобто можемо переходити до більш детального аналізу.

За поставленим завданням, ми досліджуємо зв'язок між даними в стовпці 39 Menu Eatgirvid (який являє собою сукупність відповідей на питання, яке саме масло вживають люди) та даними в інших стовпцях.

Зважаючи на те, що кореляцій буде велика кількість, то виведемо лише 10 найбільших за модулем значень:

```
Топ-10 кореляцій для стовпця '39 Menu Eatgirvid':
Menu Eatgiroil                          0.583221
Menu Termin Vagitnosti                  0.254927
Number Covidpostsindromtimeweeks        0.221860
Checkbox Eatgirtvar                     0.209363
83 misec Number Covidpostsindrommons    0.105967
17 Menu Education                       0.102353
99 Radio Addanketanew                   0.092214
29 Radio Eatovoch                       0.089921
_ Status                                0.084659
53 Radio Tiskizmer                      0.077179
Name: 39 Menu Eatgirvid, dtype: float64
```

Рис.24 Топ-10 кореляцій для 39 Menu Eatgirvid

На рис. 24 можемо побачити топ-10 кореляцій для стовпця 39 Menu Eatgirvid, які демонструють найбільш значущі взаємозв’язки із такими даними: тип віджиму масла; термін вагітності; тривалість постковідного синдрому (у тижнях); тип тваринного жиру; тривалість постковідного синдрому (у місяцях); тип освіти; вживання овочів; статус; артеріальний тиск. Кореляціями із даними в стовпцях Status можна знехтувати, оскільки містять лише слова unread.

Вважаємо, що кореляції між даними існують та можемо продовжити дослідження далі.

3.4. Мультиколінеарність

В даному розділі ми спробуємо визначити чи присутня мультиколінеарність між даними. Дана перевірка необхідна, щоб визначити чи присутні сильні кореляції між незалежними змінними. Мультиколінеарність може призвести до таких проблем:

1. Невірне оцінювання коефіцієнтів. При сильній кореляції, оцінки їхніх коефіцієнтів можуть бути неточними;
2. Складне визначення реальний вплив на залежну змінну;
3. Збільшуються стандартні помилки коефіцієнтів, що спричинить помилки в статистичних тестах.

Перевіримо чи присутня мультиколінеарність для наших даних. Для цього нам необхідно обчислити коефіцієнти VIF (*Variance Inflation Factor*) , які показують, наскільки збільшується дисперсія коефіцієнта змінної внаслідок кореляції із іншими і обчислюється за формулою:

$$VIF(X_i) = \frac{1}{1 - R_i^2}$$

де R_i^2 – коефіцієнт детермінації.

Якщо значення коефіцієнта VIF перевищує 10, то можемо впевнено сказати, що присутня мультиколінеарність. Розрахуємо даний для кожної змінної за допомогою функції variance_inflation_factor з бібліотеки statsmodels.

	Feature	VIF
1	2 Date	1030.297053
138	85 Radio Covidvakcina	47.399213
6	6 Radio Vagitnist	40.999733
4	4 Radio Stat	38.120245
10	8A Date Vidizd	31.106891
..	...	...
113	61 Menu Poglmisgrud	1.210884
127	75 Menu Bilzakrep	1.210038
101	54 Menu Pulsizmer	1.197315
51	26 Menu Eatwoter	1.187583
65	34 Menu Eatsoltwo	1.178033

[164 rows x 2 columns]

Рис. 25 Коефіцієнти VIF

Значення VIF на рис. 25 вказують на те, що мультиколінеарність присутня.

Наступним нашим кроком було б видалення змінних, які містять високі значення даного коефіцієнта (більше 10), але якщо реалізувати даний крок, то будуть видалені дані із стовпця 39 Menu Eatgirvid, який

необхідний нам для подальшого дослідження. Приймаємо рішенні про зниження порогу для коефіцієнту VIF до 50, оскільки далі ми будемо проводити факторний аналіз, який побудований на кореляціях і висока мультиколінеарність може вплинути на інтерпретацію, але це зовсім не означає, що аналіз стане некоректним, а також факторний аналіз об’єднує змінні в меншу кількість факторів, що відповідно знижує проблему із мультиколінеарністю.

Видаляємо змінні із коефіцієнтом VIF>50:

```
Видаляємо змінну: 2 Date с VIF = 1030.297053161064
Видаляємо змінну: _ Status с VIF = 184.6328537578929
Видаляємо змінну: 85 Radio Covidvakcina с VIF = 152.77511708192057
Видаляємо змінну: 6 Radio Vagitnist с VIF = 132.3990017911976
Видаляємо змінну: 44 dlina Number Dovgtila с VIF = 102.14406460579887
Видаляємо змінну: 44 massa Number Masatila с VIF = 99.94365736346526
Видаляємо змінну: 76 Radio Covid с VIF = 63.91132960405603
Видаляємо змінну: 8 Radio Misceprogiv с VIF = 50.34267214118852
Матрица VIF после снижения мультиколлинеарности:
                                Feature        VIF
149                 94 Radio Fizkult  49.428104
126        82 Radio Covidpostsindrom  40.442025
97           55A Radio Cukorizmerbig  39.748142
66                 39 Menu Eatgirvid  36.030514
52                 29 Radio Eatovoch  31.026065
..                               ...        ...
68               Checkbox Eatgirtvar   1.430708
1                      3 Date Howold   1.382899
78      45 zmensh Number Vagachengzm   1.351106
4             Menu Termin Vagitnosti   1.237458
36   21 EXTRA Textarea Badparenttext   1.226081

[156 rows x 2 columns]
```

Рис.26 Дані після видалення змінних із коефіцієнтом VIF, що більше 50

На рисунку 26 відображені нові дані із зменшеною мультиколінеарністю.

3.5. Тест Kaiser-Meyer-Olkin (KMO)

Для проведення факторного аналізу, необхідно переконатися, що дані відповідають усім необхідним вимогам. З цією метою виконуємо тест КМО та тест Бартлета на нових даних із зменшеною мультиколінеарністю.

Тест Бартлета (Bartlett's test**)** для факторного аналізу використовується для перевірки гіпотези про те, що кореляційна матриця

```
KMO тест: 0.7810240370201296
Bartlett's test: p-value = 0.0
```

Рис.27 Результати тестів КМО та Бартлетта

Дані Kaiser-Meyer-Olkin (КМО) тесту демонструють значення, яке більше за 0.6, значення $p - value$ менше за 0.05, що свідчить нам про те, що дані прийнятні для факторного аналізу.

3.6. Визначення кількості факторів

Для визначення оптимальної кількості факторів будемо використовувати метод Кайзера та Scree Plot.

Метод Кайзера (Kaiser criterion) – найбільш поширюваний підхід для визначення оптимальної кількості факторів при використанні факторного аналізу. В основі даного методу лежить використання власних значень (eigenvalues) матриці кореляцій, що і дозволяє оцінити важливість фактора.

```
Власні значення факторів: [1.25909280e+01 6.85300010e+00 6.45696701e+00 4.907
016e+00
 4.77128176e+00 4.60815134e+00 3.81688884e+00 3.24771830e+00
 2.89872181e+00 2.71791433e+00 2.67979926e+00 2.56204983e+00
 2.48008325e+00 2.44481797e+00 2.30563425e+00 2.13601786e+00
 2.04051776e+00 2.02691919e+00 2.01056231e+00 1.88856285e+00
 1.85342882e+00 1.79576117e+00 1.77941024e+00 1.72852787e+00
 1.69095122e+00 1.65864691e+00 1.59588659e+00 1.56643081e+00
 1.47032436e+00 1.45267670e+00 1.40624611e+00 1.40136587e+00
 1.37503310e+00 1.35969747e+00 1.35483611e+00 1.29866159e+00
 1.28309195e+00 1.24545921e+00 1.24313005e+00 1.21607826e+00
 1.19471373e+00 1.15894651e+00 1.14623356e+00 1.13016018e+00
 1.12130316e+00 1.09872570e+00 1.07315808e+00 1.06331180e+00
 1.04484915e+00 1.04194233e+00 1.00407293e+00 9.95575612e-01
 9.82888269e-01 9.70099137e-01 9.55873477e-01 9.32593039e-01
 9.27139061e-01 9.18495264e-01 9.09491945e-01 8.92226147e-01
 8.83173632e-01 8.77793428e-01 8.71050084e-01 8.56308597e-01
 8.46093193e-01 8.31773460e-01 8.20776570e-01 8.02689561e-01
 7.84800339e-01 7.73922043e-01 7.69416589e-01 7.49502381e-01
 7.41327996e-01 7.34487737e-01 7.26738285e-01 7.19686726e-01
 7.00728488e-01 6.86463622e-01 6.78986602e-01 6.72978960e-01
 6.64025557e-01 6.54024375e-01 6.27586804e-01 6.14236532e-01
```

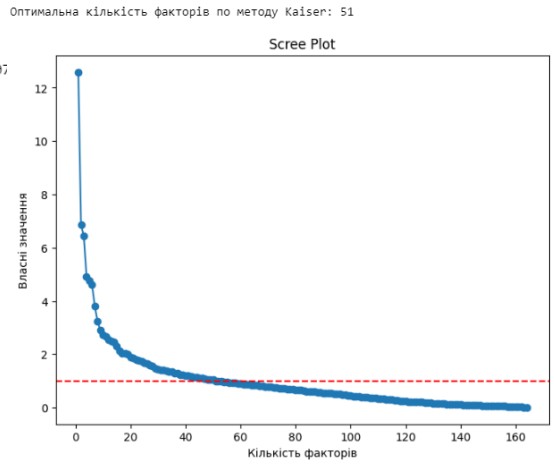


Рис. 28 Власні значення векторів та графік Scree plot

За результатами, наведеними на рис. 28, приймаємо рішення, що оптимальна кількість факторів для даного набору даних становитиме 51.

На наступному етапі буде виконано факторного аналізу із рекомендованою кількістю факторів, одночасно із зменшенням розмірності. Наша ціль полягає в тому, щоб спростити модель і зробити її більш зручною для інтерпретації.

	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6	Factor7	\
0	-0.875909	0.191411	-1.583604	-0.367624	0.049027	-0.310774	0.966282	
1	1.143784	0.013946	-1.460653	0.482180	0.675690	-0.407117	0.933653	
2	-0.875909	-0.149623	-1.615122	1.125848	-0.016529	-0.318817	0.866685	
3	-0.875909	-0.559561	-1.653665	-0.651995	0.087766	-0.367413	0.750749	
4	-0.875909	-0.054793	-1.604947	-0.630320	0.051980	2.190429	0.230557	

	Factor8	Factor9	Factor10	...	Factor42	Factor43	Factor44	Factor45
0	-0.281864	-0.326432	1.050925	...	0.089993	0.268490	2.060011	-0.330116
1	-0.303808	0.316646	1.130155	...	-0.964817	0.300080	2.257315	0.356357
2	-1.427745	0.941214	-0.568585	...	-0.321819	0.222420	1.299260	0.244741
3	-0.590738	0.364738	1.162981	...	-0.728293	-2.700252	-0.067598	1.777551
4	-0.557870	-0.086354	2.531623	...	0.322252	-1.699326	-0.813449	2.016455

	Factor46	Factor47	Factor48	Factor49	Factor50	Factor51
0	0.458316	-0.501626	-1.052536	0.042925	-0.376196	-0.328199
1	1.690038	1.057112	-2.144438	-0.239127	-0.027710	0.141331
2	-0.818950	-0.884792	-0.391015	1.446907	-0.074760	-0.928879
3	-0.363336	-0.591514	-0.524973	1.006552	0.168581	0.620474
4	-0.210368	0.228663	-0.656197	1.142891	-0.194321	0.039486

Рис.29 Факторний аналіз після зниження розмірності

На основі отриманих результатів (рис. 29), переходимо до кластеризації, яка дозволить виявити групи, які будуть об’єднувати схожі питання. Такий підхід сприятиме глибшому розумінню структури даних.

3.7. Кластеризація

3.7.1. Нормалізація факторів та кластеризація даних

Для проведення кластерного аналізу необхідно нормалізувати фактори, оскільки їх значення можуть значно відрізнятися за масштабом. Нормалізація буде забезпечувати однаковий внесок усіх факторів у результат кластеризації. Використаємо стандартну нормалізацію.

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

де z_{ij} – нормалізоване значення $i-$го спостереження за $j-$м фактором; x_{ij}- початкове значення; μ_j – середнє значення фактора j; σ_j – стандартне відхилення фактора j.

Для визначення числа кластерів будемо використовувати «Метод ліктів». Основна ідея цього методу полягає в побудові графіка залежності функції суми квадратів відстаней до центрів кластерів (Within-Cluster Sum of Squares, WCSS) від кількості факторів k.

$$WCSS = \sum_{k=1}^{k} \sum_{i \epsilon C_k} ||x_i - \mu_k||^2$$

де C_k – набір спостережень, що належать кластеру k; x_i – значення спостереження; μ_k – центр кластеру k.

Зобразимо графік і обираємо k в точці, де спостерігається різкий спад WCSS — це і є так званий "лікоть".

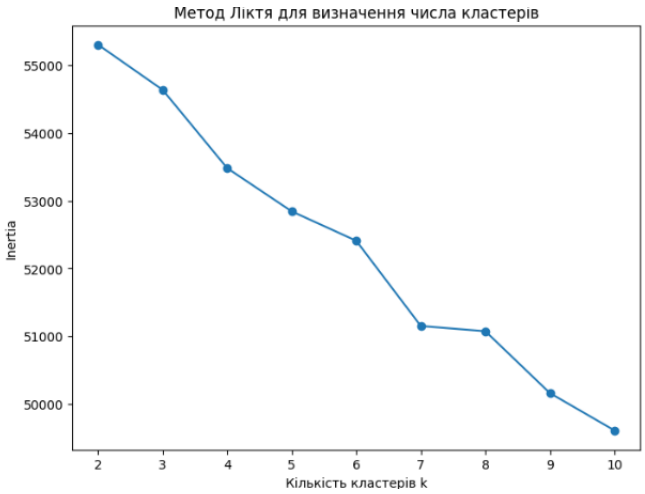


Рис. 30 Допоміжний графік для визначення числа кластерів

За графіком на рис. 30 робимо висновок, що оптимальна кількість кластерів за Методом Ліктя відповідає тій точці, де графік змінює кут нахилу, отже у діапазоні 4-6.

3.7.2. Аналіз кластерів

Створюємо нову групу даних у яких, кожен рядок буде мати номер кластера в колонці Cluster. Визначимо середні значення факторів для кожного кластера.

Cluster	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6	Factor7
0	0.209709	0.117162	-0.026657	-0.112527	-0.014872	2.261624	-0.456537
1	-0.020583	-0.046135	0.089744	-0.045212	-0.017207	-0.436832	0.078338
2	-0.027638	-0.128739	0.016542	-0.025438	0.010480	0.152058	0.209678
3	-0.056191	0.052230	0.199609	0.092108	-0.021632	-0.369274	-0.085026
4	-0.032391	0.016094	-0.193012	0.058916	0.032220	-0.411869	0.088950

Cluster	Factor8	Factor9	Factor10	...	Factor42	Factor43	Factor44
0	-0.168461	0.115127	-0.041829	...	0.044457	-0.064041	-0.002971
1	-0.197752	0.160576	0.202238	...	0.193973	-0.223899	-0.115659
2	0.075968	-0.659846	0.348356	...	-0.008191	-0.013517	-0.056042
3	0.074932	0.071463	-0.099076	...	-0.090106	0.100280	0.073912
4	0.216244	-0.016115	-0.269882	...	-0.173822	0.216678	0.104892

Cluster	Factor45	Factor46	Factor47	Factor48	Factor49	Factor50	Factor51
0	0.019800	0.055016	-0.061475	0.014867	0.014119	0.027728	0.015303
1	-0.258542	0.143761	0.125325	0.128214	0.352581	0.093453	0.282723
2	0.010202	-0.017134	0.023436	0.004582	0.014869	0.027614	0.026419
3	-0.004911	-0.250710	0.018331	-0.121752	-0.104343	-0.009404	-0.179142
4	0.262066	-0.038412	-0.122623	-0.079546	-0.328188	-0.115905	-0.220308

[5 rows x 51 columns]

Рис. 31 Середні значення факторів для кожного кластеру

3.7.3. Візуалізація кластерів

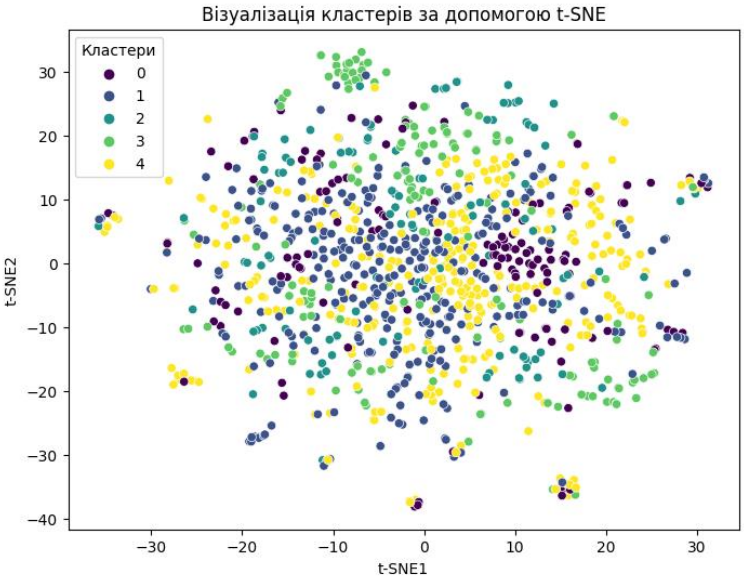


Рис. 32 Візуалізація кластерів

На рис. 32 зображено, що кластери перекривають один одного і межі між групами не є чіткими. Скоріше за все це пов’язано із тим, що фактори

недостатньо розмежовують групи. Також можемо помітити, що кластери є розсіяними, що вказує на те, що може бути присутня велика внутрішня варіативність.

3.8. Аналіз 39 Menu Eatgirvid

Оскільки наше завдання полягало в тому чи є взаємозв'язок між вживанням вершкового масла та рослинної олії, то варто визначити, які саме елементи присутні в стовпці 39 Menu Eatgirvid для подальшого аналізу.

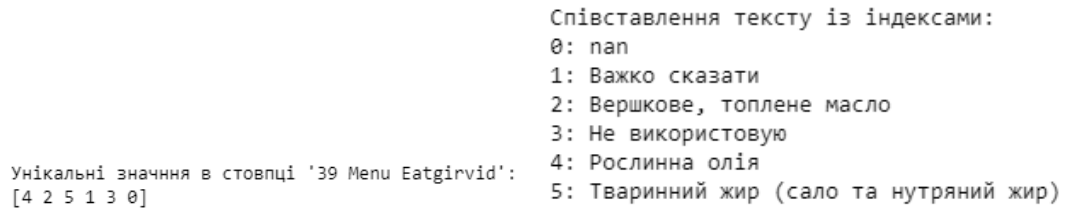


Рис. 33 Відображення даних у стовпці 39 Menu Eatgirvid

Створюємо 2 підмножини, які будуть містити значення 2 (вершкове масло) та 4 (рослинна олія) та шукаємо кореляції.

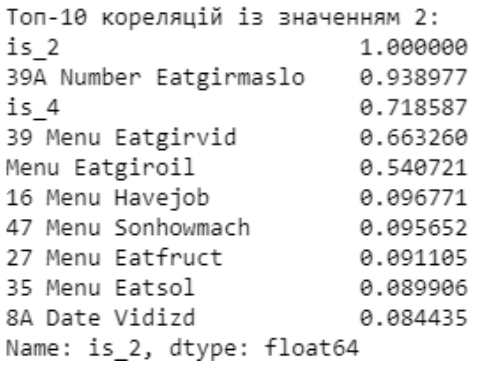


Рис.34 Відображення кореляцій із вершковим маслом та іншими даними

Отримані результати свідчать про те, що існує взаємозв’язок із відсотком жирності масла; вид жиру; наявністю в людини роботи; кількістю годин сну на добу; вживання фруктів ; вживання солі; датою.

```
Топ-10 кореляцій із значенням 4:
is_4                      1.000000
39 Menu Eatgirvid         0.786277
Menu Eatgiroil            0.744645
is_2                      0.718587
39A Number Eatgirmaslo    0.673183
Checkbox Eatgirtvar       0.224869
16 Menu Havejob           0.105819
30 Number Eatovochnumb    0.097000
4 Radio Stat              0.092692
53A Menu Tiskizmerpokaz   0.089851
Name: is_4, dtype: float64
```

Рис. 35 Відображення кореляцій із рослинною олією та іншими даними

Подані результати на рис. 35 свідчать про те, що існує кореляція між ролинною олією та видом жиру, який споживає людина; видом віджиму олії; видом жирності масла; вживанням тваринного чи рослинного жиру; наявністю роботи; вживанням свіжих овочів; статі; показанням тиску.

Оскільки ми ставили за ціль дізнатися, як саме впливають вершкове масло та рослинна олія на здоров’я, буде доцільним звернути увагу і виділити такі зв’язки: вершкове масло має кореляцію із кількістю сну на добу, а рослинна олія на показання тиску.

Висновки

Загальний аналіз дослідження дозволяє нам дійти до висновків і зрозуміти, як саме події 2022-2023 років (повномасштабна війна в Україні, пандемія COVID-19, економічна та політична криза) вплинули на психоемоційний стан населення, стан здоров'я та академічну успішність учнів і студентів. Робота складалась із трьох частин, перші дві стосувалися аналізу впливу стресових факторів на студентів, а третя частина – здоров'я пацієнтів різних вікових, соціальних категорій.

В першій частині дослідження було використано модель логістичної регресії, яка була побудована на даних анкетування 40 студентів НТТУУ «КПІ ім. Ігоря Сікорського» у лютому 2024 року. Ця модель логістичної регресії може бути використана для прогнозування рівня мотивації студентів, їх успішності. Аналіз проводився за допомогою програмного забезпечення MS Excel і пакету Real Statistics. Дані інструменти аналізу та такі методи, як покроковий відбір, використанням ROC-кривих, статистики J-Юдена дозволили нам визначити найбільш ефективну і просту модель.

Друга частина дослідження містить у собі аналіз вибірки із відповідями від 150 студентів, котрі навчалися в НТТУ «КПІ ім. Ігоря Сікорського» у травні 2024 року. На цьому етапі було застосовано факторний аналіз із застосуванням Scree plot, коваріаційної матриці, ортогональний метод обертання Varimax. В результаті було виявлено, 5 латентних факторів, які впливають на успішність і мотивацію: емоційний стрес і фізичні реакції, моральний та професійний настрій, плани на майбутнє, професійні перспективи, а також зниження життєвого інтересу та активності.

В третій частині аналіз був спрямований на те, щоб виявити зв'язки між вживанням певних харчових продуктів та станом здоров'я пацієнтів. Дані

були результатом опитування 1101 пацієнта, яким було задано більше 150 питань. Результатом цієї частини стало те, що було знайдено кореляції між вживанням рослинної олії та рівнем артеріального тиску; вершкового масла і тривалістю сну.

Результати, які були отримані в даному дослідженні, мають велике практичне значення і можуть бути використані для зміни та вдосконалення освітніх програм, для підвищення мотивації студентів, а також розробити більш ефективні системи харчування для пацієнтів із урахуванням кореляцій між показниками стану їхнього здоров’я та вживаними продуктами.

Використана література

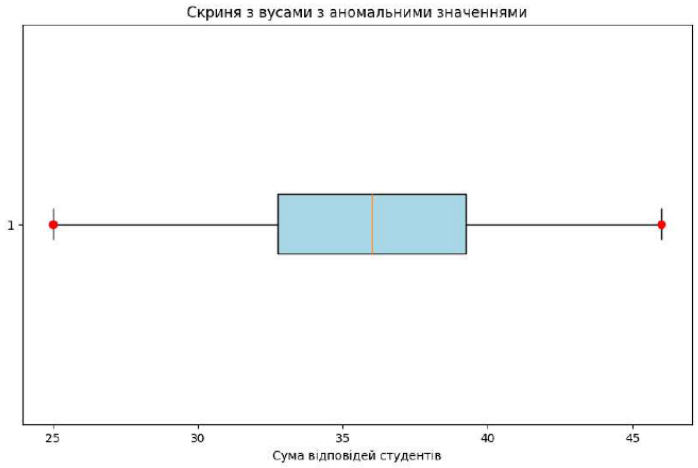
1. Гур'янов, В. Г., Лях, Ю. Є., Парій, В. Д., Короткий, О. В., Чалий, О. В., Чалий, К. О., Цехмістер, Я. В. **Посібник з біостатистики. Аналіз результатів медичних досліджень у пакеті EZR (R–statistics): навчальний посібник.** – Київ: Вістка, 2018. – 208 с
2. Johnson, R. A., Wichern, D. W. **Applied Multivariate Statistical Analysis.** – 6th Ed. – Pearson, 2007.
3. Howell, D. C. **Statistical Methods for Psychology.** – Wadsworth, Cengage Learning, University of Vermont, 2010, 2007. – 780 с.
4. McDonald, J. H. **Handbook of Biological Statistics.** – 3rd Ed. – Sparky House Publishing, Baltimore, Maryland, 2014.
5. Rencher, A. C., Christensen, W. F. **Methods of Multivariate Analysis.** – 3rd Ed. – Wiley, 2012.
6. Nield, T. **Essential Math for Data Science.** – 2025.
7. Real Statistics Resource Pack (Release 8.9.1). URL: https://www.real-statistics.com
8. ROC curve analysis. URL: https://www.medcalc.org/manual/roc-curves.php
9. URL: https://habr.com/ru/articles/687338/

Додаток 1

In [65]:
```
from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt
from factor_analyzer import FactorAnalyzer
from factor_analyzer.factor_analyzer import calculate_kmo
from scipy.stats import bartlett
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np

file_path = r"C:\Users\helen\OneDrive\Рабочий стол\
Магистерская дисертация\Students' Answers.xlsx"
df = pd.read_excel(file_path)

# Визначення аномальних значень
distribution = df.sum(axis=1)
z_scores = stats.zscore(distribution)
threshold = 2

outliers = distribution[(z_scores < -threshold) | (z_scores > threshold)]
print("Аномальні дані:")
print(outliers)
```

```
Аномальні дані:
1     46
21    25
22    25
dtype: int64
```

In [66]:
```
plt.figure(figsize=(10, 6))
plt.boxplot(distribution, vert=False, patch_artist=True,
            boxprops=dict(facecolor="lightblue"))

for index in outliers.index:
    plt.plot(outliers[index], 1, 'ro')

plt.xlabel("Сума відповідей студентів")
plt.title("Скриня з вусами з аномальними значеннями")
plt.show()
```



In [67]:
```
df_cleaned = df.drop(outliers.index)

print("Дані без аномалій:")
print(df_cleaned)
```

Дані без аномалій:

	X1	X2	X4	X6	X7	X8	X9	X10	X11	X12	Y14
0	5	2	2	4	5	5	4	3	3	4	4
2	4	1	1	2	5	4	3	4	3	2	4
3	5	5	1	5	5	5	5	5	1	2	3
4	5	3	5	5	5	3	1	4	3	4	3
5	4	1	2	3	4	4	3	2	4	4	4
6	4	2	1	4	5	4	4	5	5	4	4
7	5	2	4	3	4	2	2	3	3	4	4
8	4	2	4	4	5	2	4	1	2	3	2
9	2	4	4	2	5	2	1	1	3	3	4
10	4	4	3	4	4	2	2	3	4	3	3
11	5	1	1	5	5	5	5	5	1	2	3
12	5	2	3	5	5	5	2	4	3	3	3
13	5	3	3	4	5	5	5	5	1	2	2
14	4	4	3	4	4	4	4	3	3	2	2
15	5	3	1	4	5	4	2	1	2	4	1
16	4	2	1	4	5	3	4	5	3	3	2
17	3	2	1	4	3	2	3	4	3	3	4
18	4	3	1	4	4	5	4	3	3	1	5
19	4	4	1	5	4	3	4	4	2	1	2
20	4	5	5	3	5	2	4	5	3	4	4
23	3	4	4	3	4	4	3	4	2	3	2
24	4	5	2	5	5	5	4	1	1	2	3
25	4	3	5	4	5	4	4	4	2	4	3
26	5	2	4	3	5	4	3	4	3	4	2
27	5	4	2	3	5	4	3	4	3	4	3
28	4	4	2	4	5	4	4	3	3	4	2
29	4	2	5	3	5	2	1	2	3	4	3
30	4	5	5	5	5	3	1	5	1	1	1
31	4	2	3	3	5	2	2	2	2	3	2
32	4	3	1	1	5	3	3	1	3	3	3
33	4	2	2	2	3	1	1	1	4	4	2
34	5	3	3	4	5	3	5	3	3	3	2
35	5	2	1	4	5	2	2	4	2	4	3
36	4	2	2	3	5	4	4	4	2	2	5
37	4	3	2	1	5	2	5	1	2	5	4
38	4	3	3	4	3	2	2	2	3	3	2
39	3	1	1	3	5	2	2	1	3	4	4

In [68]:
```
data = df_cleaned
correlation_matrix = data.corr()
print("Кореляційна матриця на нових даних:")
print(correlation_matrix)
```

```
Кореляційна матриця на нових даних:
           X1        X2        X4        X6        X7        X8        X9  \
X1   1.000000 -0.095393 -0.049594  0.365805  0.279206  0.394166  0.223224
X2  -0.095393  1.000000  0.282160  0.231267  0.032866  0.093111  0.102456
X4  -0.049594  0.282160  1.000000  0.020249  0.093137 -0.242901 -0.335648
X6   0.365805  0.231267  0.020249  1.000000  0.030028  0.422959  0.154376
X7   0.279206  0.032866  0.093137  0.030028  1.000000  0.324438  0.218460
X8   0.394166  0.093111 -0.242901  0.422959  0.324438  1.000000  0.519016
X9   0.223224  0.102456 -0.335648  0.154376  0.218460  0.519016  1.000000
X10  0.332567  0.110594  0.044706  0.471951  0.145236  0.395339  0.307485
X11 -0.186788 -0.310003 -0.051313 -0.329406 -0.328685 -0.297039 -0.291976
X12  0.087053 -0.282103  0.168545 -0.417736  0.103303 -0.367239 -0.220533
Y14 -0.217220 -0.280279 -0.207340 -0.299947  0.012713  0.058604  0.130842

          X10       X11       X12       Y14
X1   0.332567 -0.186788  0.087053 -0.217220
X2   0.110594 -0.310003 -0.282103 -0.280279
X4   0.044706 -0.051313  0.168545 -0.207340
X6   0.471951 -0.329406 -0.417736 -0.299947
X7   0.145236 -0.328685  0.103303  0.012713
X8   0.395339 -0.297039 -0.367239  0.058604
X9   0.307485 -0.291976 -0.220533  0.130842
X10  1.000000 -0.148799 -0.294099  0.005093
X11 -0.148799  1.000000  0.456789  0.300314
X12 -0.294099  0.456789  1.000000  0.111749
Y14  0.005093  0.300314  0.111749  1.000000
```

In [69]:
```
#Kaiser-Meyer-Olkin test
kmo_all, kmo_model = calculate_kmo(data)
print("KMO Model Score:", kmo_model)
```

```
KMO Model Score: 0.6302152990566174
```

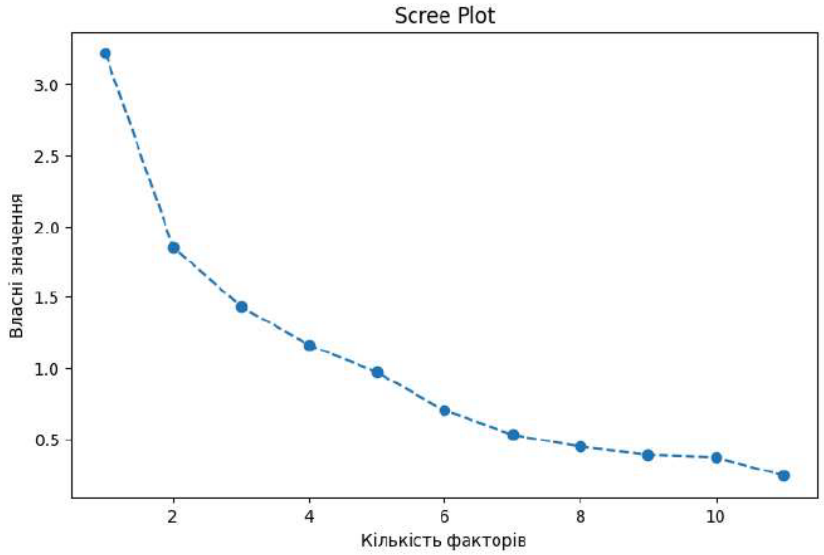
In [70]:
```
# Bartlett's test
bartlett_result = bartlett(*[data[col] for col in data.columns])
print("Bartlett's test:", bartlett_result)
```

```
Bartlett's test: BartlettResult(statistic=43.9894948721679, pvalue=3.3059972970
513196e-06)
```

In [71]:
```
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
pca = PCA()
pca.fit(data_scaled)
explained_variance = pca.explained_variance_
explained_variance_ratio = pca.explained_variance_ratio_
```

In [72]:
```
#Scree plot
plt.figure(figsize=(8, 5))
plt.plot(range(1, len(explained_variance) + 1), explained_variance,
         marker='o', linestyle='--')
plt.title('Scree Plot')
plt.xlabel('Кількість факторів')
plt.ylabel('Власні значення')
plt.show()

# Визначення кількості факторів
n_factors = np.sum(explained_variance > 1)
print(f'Рекомендована кількість факторів: {n_factors}')

# PCA з рекомендованою кількістю факторів
```

```
pca = PCA(n_components=n_factors)
data_pca = pca.fit_transform(data_scaled)
```




```
Рекомендована кількість факторів: 4
```

In [93]:
```
# Відображення факторних навантажень
fa = FactorAnalyzer(n_factors=4, rotation=None)
fa.fit(data_scaled)
factor_loadings = fa.loadings_
print(factor_loadings)
```

```
[[ 0.47212527  0.23877279  0.49547747  0.24840236]
 [ 0.28695093 -0.46072117 -0.02678743 -0.19639567]
 [-0.10696658 -0.46314949  0.36574769 -0.08537679]
 [ 0.69170542 -0.24458791  0.05805324  0.40803559]
 [ 0.3197193   0.21227518  0.41065339 -0.41996473]
 [ 0.72363702  0.34680363 -0.04837743 -0.01924569]
 [ 0.5207001   0.38983654 -0.15413739 -0.18601868]
 [ 0.52909418  0.0667976   0.04928119  0.19420782]
 [-0.59351176  0.25937787 -0.03232283  0.36466347]
 [-0.58591511  0.28192993  0.57867759  0.01327416]
 [-0.2080326   0.47546769 -0.24273643 -0.06834998]]
```

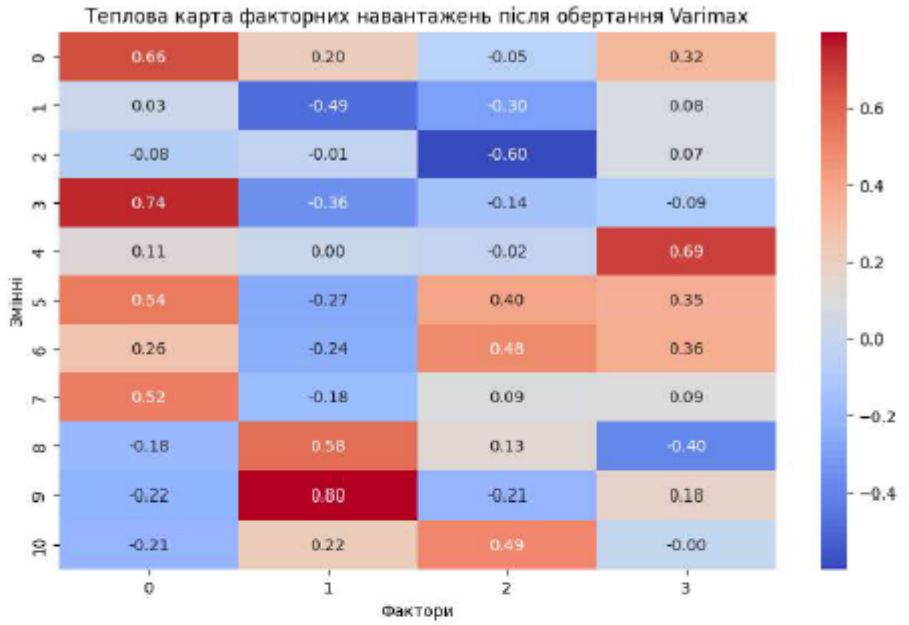
In [115…
```
# Відображення факторних навантажень (Varimax)

from factor_analyzer import FactorAnalyzer
from sklearn.preprocessing import StandardScaler
fa = FactorAnalyzer(n_factors=4, rotation="varimax")  # 3 фактори
fa.fit(data_scaled)
factor_loadings_varimax = fa.loadings_
print(factor_loadings_varimax)
```

```
[[ 0.6628342   0.19957306 -0.05239965  0.32433327]
 [ 0.02711947 -0.485951   -0.300407    0.08225239]
 [-0.07882671 -0.01413641 -0.59638867  0.07012093]
 [ 0.74236243 -0.35766679 -0.14355836 -0.09222985]
 [ 0.11227879  0.00320074 -0.02329873  0.69219084]
 [ 0.53780763 -0.26837777  0.40258909  0.35113021]
 [ 0.26256252 -0.24145114  0.47691573  0.35605626]
 [ 0.5249153  -0.17838285  0.09036863  0.09499941]
 [-0.18465612  0.5829229   0.13016764 -0.40338158]
 [-0.22078354  0.79669217 -0.20811449  0.17620877]
 [-0.20823312  0.215642    0.49301668 -0.00344038]]
```

In [94]:
```
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.heatmap(factor_loadings, annot=True, cmap="coolwarm",
            fmt='.2f', cbar=True)
plt.title("Теплова карта стандартних факторних навантажень")
plt.xlabel("Фактори")
plt.ylabel("Змінні")
plt.show()
```

n [116…
```
plt.figure(figsize=(10, 6))
sns.heatmap(factor_loadings_varimax, annot=True,
            cmap="coolwarm", fmt='.2f', cbar=True)
plt.title("Теплова карта факторних навантажень після обертання Varimax")
plt.xlabel("Фактори")
plt.ylabel("Змінні")
plt.show()
```

In [116…
```
plt.figure(figsize=(10, 6))
sns.heatmap(factor_loadings_varimax, annot=True,
            cmap="coolwarm", fmt='.2f', cbar=True)
plt.title("Теплова карта факторних навантажень після обертання Varimax")
plt.xlabel("Фактори")
plt.ylabel("Змінні")
plt.show()
```



In [117…

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

factor_loadings_df = pd.DataFrame(factor_loadings, index=data.columns,
                                  columns=[f'Factor{i+1}' for i in range(factor_
factor_loadings_varimax_df = pd.DataFrame(factor_loadings_varimax, index=data.co
                                          columns=[f'Factor{i+1}' for i in range

ax = factor_loadings_df.plot(kind='bar', figsize=(12, 6), width=0.4, color='gree
factor_loadings_varimax_df.plot(kind='bar', ax=ax, width=0.4, color='red', legen

legend = plt.legend(["Без обертання", "Після обертання"], loc="upper left")

legend.get_texts()[0].set_color("green")
legend.get_texts()[1].set_color("red")


plt.title("Порівняння факторних навантажень (без обертання vs після обертання Be
plt.xlabel("Змінні")
plt.ylabel("Навантаження")

plt.show()
```

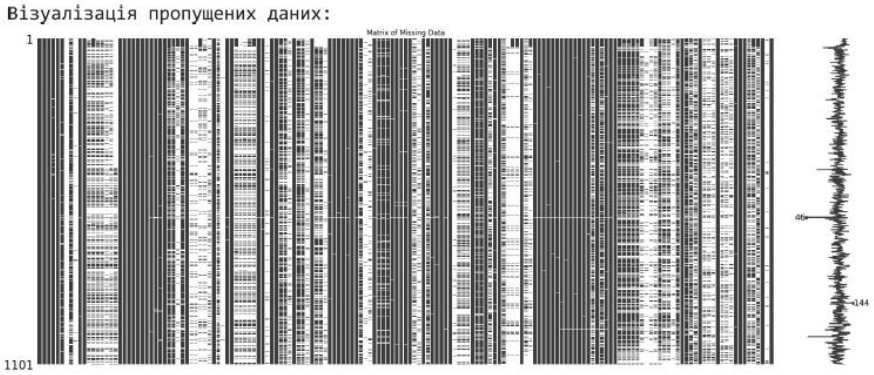

Порівняння факторних навантажень (без обертання vs після обертання Варимакс)
Без обертання
Після обертання
Навантаження
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
X1
X2
X4
X6
X7
X8
X9
X10
X11
X12
Y14
Змінні

In [ ]:

Додаток 2

In [2]:
```
import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt

file_path = r"C:\Users\helen\OneDrive\Рабочий стол\data_20.03.2024 data med.xlsx
df = pd.read_excel(file_path)

print("Візуалізація пропущених даних:")
msno.matrix(df)
plt.title("Matrix of Missing Data")
plt.show()

msno.bar(df)
plt.title("Bar Plot of Missing Data")
plt.show()
```

Візуалізація пропущених даних:




Bar Plot of Missing Data

In [40]:

```
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import FactorAnalysis
import matplotlib.pyplot as plt
import seaborn as sns

file_path = r"C:\Users\helen\OneDrive\Рабочий стол\data_20.03.2024 data med.xlsx
df = pd.read_excel(file_path)
print("Перші рядки данних:")
print(df.head())

#Процес очистки даних
#Видаляємо ті, в яких пропущені значення
df_cleaned = df.dropna(axis=1, how='all').copy()

# Перетворюємо категоріальні стовпці в числові:
categorical_columns = df_cleaned.select_dtypes(include=['object', 'datetime']).c
label_encoder = LabelEncoder()

for col in categorical_columns:
    print(f"Опрацювання стовпця: {col}")

    # Перетворимо в числа, ті стовпці, які мають дати (timestamp)
    if pd.api.types.is_datetime64_any_dtype(df_cleaned[col]):
        print(f"Стовпець {col} містить дати. Перетворюємо в timestamp.")
        df_cleaned[col] = df_cleaned[col].astype('int64') / 10**9  # Перетворенн

    # Перетворимо ті стовпці, які містять текст
    else:
        # Заповнення пропусков найбільш частим значенням:
        imputer = SimpleImputer(strategy='most_frequent')
        df_cleaned[col] = imputer.fit_transform(df_cleaned[[col]].astype(str))

        # Кодуємо значення за допомогою LabelEncoder
        df_cleaned[col] = label_encoder.fit_transform(df_cleaned[col])

# Необхідно запевнитися, що всі стовпці мають числовий тип
print("Типи данних після перетворення:")
print(df_cleaned.dtypes)

# Заповнення пропущених значень в числових стовпцях середнім значенням:
numeric_columns = df_cleaned.select_dtypes(include=['float64', 'int64']).columns
imputer = SimpleImputer(strategy='mean')
```

```
df_imputed = pd.DataFrame(imputer.fit_transform(df_cleaned[numeric_columns]), co

# Масштабування даних
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_imputed)

# Перетворимо numpy.ndarray назад в DataFrame
df_scaled = pd.DataFrame(df_scaled, columns=df_imputed.columns)

# Перевіримо результати:
print("Перші рядки після маштабування:")
print(df_scaled.head())


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

```
Перші рядки данних:
   Id              2 Date _ Status 3 Date Howold 4 Radio Stat  \
0   1 2023-01-08 16:07:37   unread    1975-02-15       Жіноча
1   2 2023-01-08 16:23:59   unread    1981-07-04       Жіноча
2   3 2023-01-08 18:37:36   unread    1988-06-26     Чоловіча
3   4 2023-01-08 20:05:37   unread    2004-07-18       Жіноча
4   5 2023-01-08 21:14:23   unread    1984-10-08       Жіноча

                     5 Menu Simstan  5A Menu Simstan2 6 Radio Vagitnist  \
0  Перебуваю у зареєстрованому шлюбі               NaN               Ні
1  Перебуваю у зареєстрованому шлюбі               NaN               Ні
2  Перебуваю у зареєстрованому шлюбі               NaN              NaN
3            Ніколи не була у шлюбі                NaN               Ні
4  Перебуваю у зареєстрованому шлюбі               NaN               Ні

   Menu Termin Vagitnosti                  7 Checkbox 118  ...  \
0                     NaN  Маю дітей віком понад 18 років  ...
1                     NaN  Маю дітей віком понад 18 років  ...
2                     NaN       Маю дітей віком 6-9 років  ...
3                     NaN                    Дітей не маю  ...
4                     NaN      Маю дітей віком до 6 років  ...

  91B Number Sportserednm 92 Number Sidlegvdenh  \
0                    50.0                  14.0
1                     NaN                   3.0
2                     NaN                   5.0
3                     NaN                   7.0
4                    10.0                   8.0

                                93 Menu Typedrive 94 Radio Fizkult  \
0  Метро/тролейбус/автобус/маршрутне таксі/трамвай              Так
1                                Автомобіль/таксі              Так
2  Метро/тролейбус/автобус/маршрутне таксі/трамвай               Ні
3                                Автомобіль/таксі              Так
4                                     Ходжу пішки               Ні

  95 Radio Sportprof       96 Menu Fizkultwhy  97 Menu Whotfizkult  \
0                 Ні  Для покращення здоров'я          Брак часу
1                 Ні  Для покращення здоров'я          Брак часу
2                NaN  Для покращення здоров'я          Брак часу
3                 Ні  Для покращення здоров'я          Брак часу
4                NaN  Для покращення здоров'я  Відсутність бажання

  Radio Familyanketa        98 Textarea Familyanketa 99 Radio Addanketanew
0                 Ні                             NaN                   Так
1                Так  Кармацьких Анатолій Вікторович                   Так
2                 Ні                             NaN                   Так
3                 Ні                             NaN                   Так
4                 Ні                             NaN                   Так

[5 rows x 165 columns]
Опрацювання стовпця: 2 Date
Стовпець 2 Date містить дати. Перетворюємо в timestamp.
Опрацювання стовпця:   Status
Опрацювання стовпця: 3 Date Howold
Стовпець 3 Date Howold містить дати. Перетворюємо в timestamp.
Опрацювання стовпця: 4 Radio Stat
Опрацювання стовпця: 5 Menu Simstan
Опрацювання стовпця: 6 Radio Vagitnist
Опрацювання стовпця: 7 Checkbox 118
```

Опрацювання стовпця: 8 Radio Misceprogiv
Опрацювання стовпця: 8A Date Vidizd
Стовпець 8A Date Vidizd містить дати. Перетворюємо в timestamp.
Опрацювання стовпця: 8B Menu Oblasti
Опрацювання стовпця: 8V Radio Leavukr
Опрацювання стовпця: 8G Radio Backhome
Опрацювання стовпця: 8B misto Text Town R Now
Опрацювання стовпця: 8D Menu Oblasti Now
Опрацювання стовпця: Radio Status Vpo
Опрацювання стовпця: 9 Radio Remembplace
Опрацювання стовпця: 10 Radio Reminteres
Опрацювання стовпця: 11 Radio Pochdalek
Опрацювання стовпця: 12 Radio Remlove
Опрацювання стовпця: 13 Radio Remfuture
Опрацювання стовпця: 14 Radio Remson
Опрацювання стовпця: 15 Radio Adddratl
Опрацювання стовпця: 16 Menu Havejob
Опрацювання стовпця: 17 Menu Education
Опрацювання стовпця: 18 Menu Workgroup
Опрацювання стовпця: 19 Radio Alkohol
Опрацювання стовпця: 19A Radio Canstopalkogol
Опрацювання стовпця: 19B Menu Helpalkogol
Опрацювання стовпця: 19V Menu Noalkogol
Опрацювання стовпця: 19D Menu Alkostartwar
Опрацювання стовпця: 20 Radio Smock
Опрацювання стовпця: 20A Radio Hminismoke
Опрацювання стовпця: 20B Radio Docminismoke
Опрацювання стовпця: 20V Menu Whstopsmok
Опрацювання стовпця: 20G Radio Electrsmock
Опрацювання стовпця: 20D Radio Hminismoket
Опрацювання стовпця: 21 Radio Badparent
Опрацювання стовпця: 21 list Checkbox Badparentcheck
Опрацювання стовпця: 21 EXTRA Textarea Badparenttext
Опрацювання стовпця: 22 Menu Eat
Опрацювання стовпця: 23 Radio Eatchange
Опрацювання стовпця: 23A Menu Eathow
Опрацювання стовпця: 23B Radio Eatfinchange
Опрацювання стовпця: 23V Checkbox Eathowprod
Опрацювання стовпця: 23G Checkbox Eatdef
Опрацювання стовпця: 23D Checkbox Eatzapas
Опрацювання стовпця: 24 Menu Eatapetit
Опрацювання стовпця: 25 Menu Eatqwont
Опрацювання стовпця: 25 EXTRA Text Eatqwontvar
Опрацювання стовпця: 26 Menu Eatwoter
Опрацювання стовпця: 27 Menu Eatfruct
Опрацювання стовпця: 28 dniv Radio Eatfructdn
Опрацювання стовпця: 29 Radio Eatovoch
Опрацювання стовпця: 30 dniv Radio Eatovochdn
Опрацювання стовпця: 31 Radio Eatovochterm
Опрацювання стовпця: 32dniv Radio Eatovochdnkar
Опрацювання стовпця: 33 Menu Eatcartop
Опрацювання стовпця: 34 Menu Eatsoltwo
Опрацювання стовпця: 35 Menu Eatsol
Опрацювання стовпця: 36 Menu Eatsolbig
Опрацювання стовпця: 37 Menu Eatsolprod
Опрацювання стовпця: 38 Radio Sollive
Опрацювання стовпця: 39 Menu Eatgirvid
Опрацювання стовпця: Menu Eatgiroil
Опрацювання стовпця: Checkbox Eatgirtvar
Опрацювання стовпця: 40 Menu Eatsolod

Опрацювання стовпця: 40A Radio Eatsolodman
Опрацювання стовпця: 40B Checkbox Eatsolodprod
Опрацювання стовпця: 40V Checkbox Eatsolodwhy
Опрацювання стовпця: 41 Menu Eatnothome
Опрацювання стовпця: 45 Radio Vagacheng
Опрацювання стовпця: 46 Radio Soncheng
Опрацювання стовпця: 46A Menu Sonchenghow
Опрацювання стовпця: 47 Menu Sonhowmach
Опрацювання стовпця: 48 Menu Sonharak
Опрацювання стовпця: 49 Radio Sonafter
Опрацювання стовпця: 50 Menu Vtoma
Опрацювання стовпця: 51 Radio Hvorob
Опрацювання стовпця: 51 zahv Textarea Allhvorob
Опрацювання стовпця: 51A Radio Hvorobpogir
Опрацювання стовпця: 51B Radio Hvorobdefic
Опрацювання стовпця: 51V Radio Hvorobprobl
Опрацювання стовпця: 52 Radio Tisk
Опрацювання стовпця: 52A Radio Tiskbig
Опрацювання стовпця: 52B Radio Tiskliki
Опрацювання стовпця: 53 Radio Tiskizmer
Опрацювання стовпця: 53A Menu Tiskizmerpokaz
Опрацювання стовпця: 54 Menu Pulsizmer
Опрацювання стовпця: 55 Radio Cukorizmer
Опрацювання стовпця: 55A Radio Cukorizmerbig
Опрацювання стовпця: 56 Radio Cukorizmerbigdoc
Опрацювання стовпця: 56A Menu Cukorminpor
Опрацювання стовпця: 56B Menu Cukorbigpor
Опрацювання стовпця: 57 Radio Holesterinizm
Опрацювання стовпця: 57A Radio Holesterinbig
Опрацювання стовпця: 57B Menu Holesterinporad
Опрацювання стовпця: 58 Menu Pechiya
Опрацювання стовпця: 59 Menu Zdutya
Опрацювання стовпця: 60 Menu Tyaggiv
Опрацювання стовпця: 61 Menu Poglmisgrud
Опрацювання стовпця: 62 Menu Slabpisligi
Опрацювання стовпця: 63 Menu Pechiyapisligi
Опрацювання стовпця: 64 Menu Nezvichpisligi
Опрацювання стовпця: 65 Menu Perepovpisligi
Опрацювання стовпця: 66 Menu Zastraganpisligi
Опрацювання стовпця: 67 Menu Girkotapisligi
Опрацювання стовпця: 68 Menu Vidrigkapisligi
Опрацювання стовпця: 69 Menu Pechiyanahil
Опрацювання стовпця: 70 Menu Nudota
Опрацювання стовпця: 71 Menu Nudotaveare
Опрацювання стовпця: 72 Menu Bilgevit
Опрацювання стовпця: 73 Menu Bilgevitvere
Опрацювання стовпця: 74 Menu Bildefik
Опрацювання стовпця: 75 Menu Bilzakrep
Опрацювання стовпця: 76 Radio Covid
Опрацювання стовпця: 77 Menu Covidgroup
Опрацювання стовпця: 78 Radio Coviddopom
Опрацювання стовпця: 80 Radio Covidusklad
Опрацювання стовпця: 81 Menu Covidhronusklad
Опрацювання стовпця: 82 Radio Covidpostsindrom
Опрацювання стовпця: 83 Menu Covidpostsindromtime
Опрацювання стовпця: 84 Menu Covidpostsindromhow
Опрацювання стовпця: 85 Radio Covidvakcina
Опрацювання стовпця: 85B Radio Covidvakcinausklad
Опрацювання стовпця: 85 uskl Textarea Covidvakcinausklad
Опрацювання стовпця: 86 Radio Coviddouble

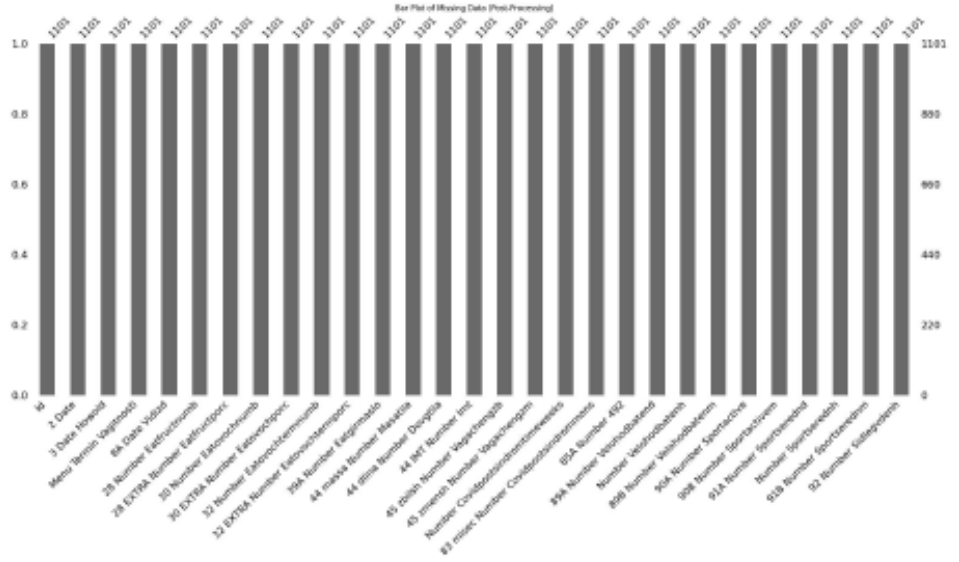
```
Опрацювання стовпця: 89 Radio Velohodbaten
Опрацювання стовпця: 90 Radio Sportactive
Опрацювання стовпця: Number Sportactiveh
Опрацювання стовпця: 91 Radio Sportseredn
Опрацювання стовпця: 93 Menu Typedrive
Опрацювання стовпця: 94 Radio Fizkult
Опрацювання стовпця: 95 Radio Sportprof
Опрацювання стовпця: 96 Menu Fizkultwhy
Опрацювання стовпця: 97 Menu Whotfizkult
Опрацювання стовпця: Radio Familyanketa
Опрацювання стовпця: 98 Textarea Familyanketa
Опрацювання стовпця: 99 Radio Addanketanew
Типи данних після перетворення:
Id                            int64
2 Date                      float64
_ Status                      int32
3 Date Howold               float64
4 Radio Stat                  int32
                             ...
96 Menu Fizkultwhy            int32
97 Menu Whotfizkult           int32
Radio Familyanketa            int32
98 Textarea Familyanketa      int32
99 Radio Addanketanew         int32
Length: 164, dtype: object
Перші рядки після маштабування:
         Id    2 Date  3 Date Howold  Menu Termin Vagitnosti  8A Date Vidizd  \
0 -1.730478 -1.530651      -0.182238            3.123655e-15       -0.875909
1 -1.727332 -1.530536      -0.019314            3.123655e-15        1.143784
2 -1.724186 -1.529593       0.158846            3.123655e-15       -0.875909
3 -1.721039 -1.528971       0.568845            3.123655e-15       -0.875909
4 -1.717893 -1.528486       0.064000            3.123655e-15       -0.875909

   28 Number Eatfructnumb  28 EXTRA Number Eatfructporc  \
0               -1.161931                 -9.537123e-01
1               -0.574687                  2.572210e-16
2               -1.161931                 -6.641077e-01
3                1.187046                 -6.641077e-01
4                0.012557                 -9.537123e-01

   30 Number Eatovochnumb  30 EXTRA Number Eatovochporc  \
0                0.000000                      0.000000
1               -0.845131                     -0.833308
2                0.000000                      0.000000
3                0.390695                     -0.833308
4               -0.227218                     -1.063863

   32 Number Eatovochtermnumb  ...  85A Number 492  89A Number Velohodbatend  \
0               -5.850645e-16  ...        0.000000                 -1.274468
1               -5.850645e-16  ...        1.430920                 -0.574153
2               -5.850645e-16  ...       -2.849815                  0.126163
3               -5.850645e-16  ...        0.000000                 -0.574153
4               -5.850645e-16  ...       -0.709448                  0.826478

   Number Velohodbatenh  89B Number Velohodbatenm  90A Number Sportactive  \
0             -1.347541                 -0.279669            1.103775e-01
1              0.000000                  0.710543           -1.682584e+00
2             -0.478682                  0.000000           -3.981174e-16
3             -0.478682                  0.710543           -3.981174e-16
4              0.000000                  2.690965           -3.981174e-16
```

```
   90B Number Sportactivem  91A Number Sportseredndd  Number Sportserednh  \
0             -3.593652e+00              1.731269e-01            -0.329866
1             -5.966286e-16             -3.686480e-16             0.000000
2             -5.966286e-16             -3.686480e-16             0.000000
3             -5.966286e-16             -1.487116e+00             0.420064
4             -5.966286e-16              1.833370e+00            -1.079796

   91B Number Sportserednm  92 Number Sidlegvdenh
0                 3.944503              1.433173
1                 0.000000             -1.172911
2                 0.000000             -0.699077
3                 0.000000             -0.225244
4                -2.052106              0.011673

[5 rows x 29 columns]
```

In [41]:

```
import missingno as msno
import matplotlib.pyplot as plt

# Візуалізація пропущених даних у обробленому DataFrame
print("Візуалізація пропущених даних після обробки:")
msno.matrix(df_scaled)  # df_scaled - це ваш оброблений DataFrame
plt.title("Matrix of Missing Data (Post-Processing)")
plt.show()

# Барплот пропущених даних
msno.bar(df_scaled)
plt.title("Bar Plot of Missing Data (Post-Processing)")
plt.show()
```

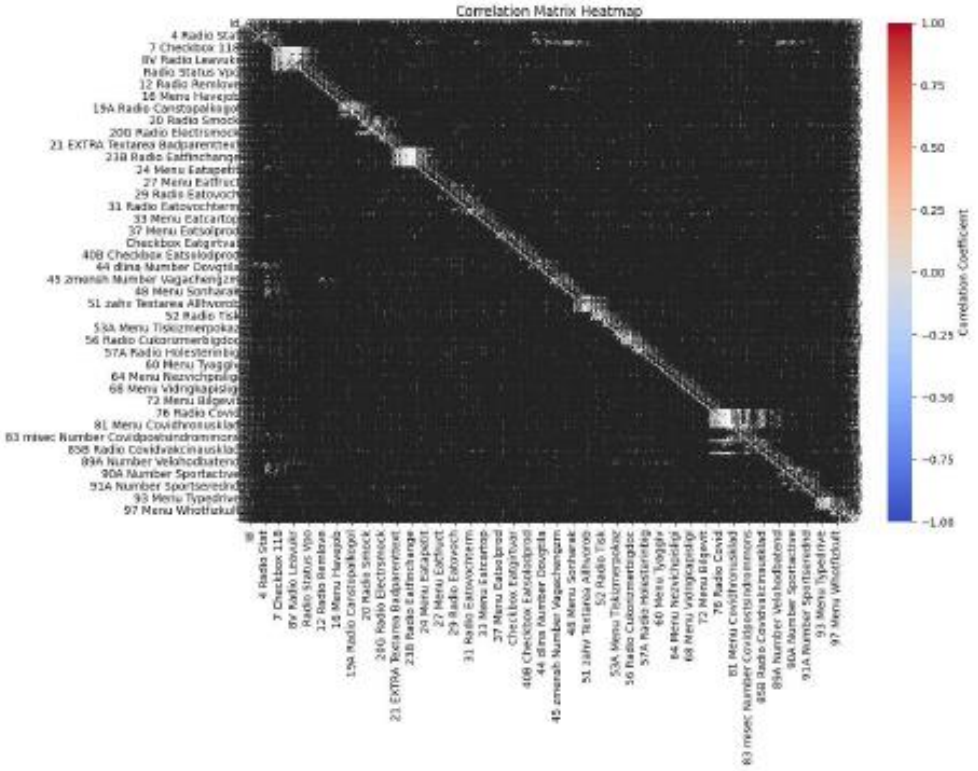
```
Візуалізація пропущених даних після обробки:
```

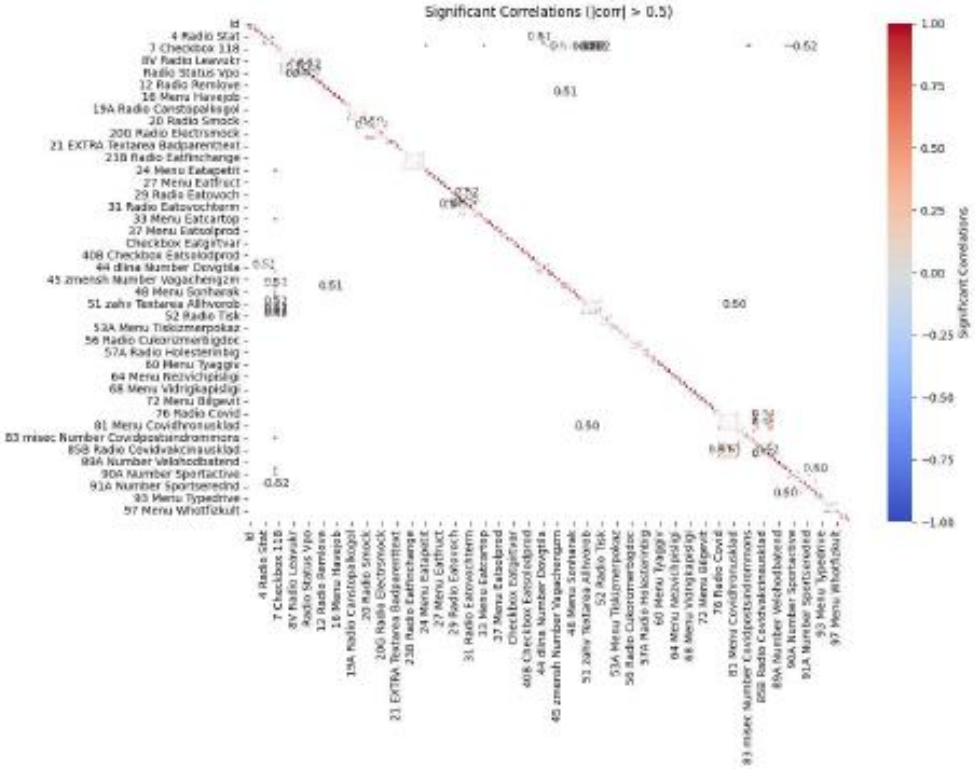


In [44]:
```python
# Переконуємося, що df_cleaned визначений і містить дані
if 'df_cleaned' not in locals():
    print("Помилка: df_cleaned не визначений. Перевірте етапи попередньої обробк
)
else:
# Перевіряємо, чи є в df_cleaned числові стовпці для обчислення кореляцій
    if df_cleaned.select_dtypes(include=['number']).empty:
        print("Помилка: у df_cleaned немає числових стовпців для обчислення коре
    else:
        # Обчислення кореляційної матриці
        correlation_matrix = df_cleaned.corr()
        print("Корреляционная матрица успешно рассчитана.")

        # Візуалізація
        plt.figure(figsize=(12, 8))
        sns.heatmap(
            correlation_matrix,
            annot=True,
            fmt=".2f",
            cmap="coolwarm",
            vmin=-1, vmax=1,
            cbar_kws={'label': 'Correlation Coefficient'}
        )
        plt.title("Correlation Matrix Heatmap")
        plt.show()

# (|кореляція| > 0.5)
significant_mask = correlation_matrix.abs() > 0.5

plt.figure(figsize=(12, 8))
sns.heatmap(
    correlation_matrix.where(significant_mask),
    annot=True,
    fmt=".2f",
    cmap="coolwarm",
    vmin=-1, vmax=1,
    cbar_kws={'label': 'Significant Correlations'},
    mask=~significant_mask  # маскуємо незначимі значення
)
```

```
plt.title("Significant Correlations (|corr| > 0.5)")
plt.show()
```

```
Корреляционная матрица успешно рассчитана.
```





In [43]:
```python
# виводимо Топ-10 найбільших кореляцій
menu_column = '39 Menu Eatgirvid'

# Матриця кореляції:
correlation_matrix = df_cleaned.corr()

# проведемо аналіз отриманої матриці кореляції
if menu_column in correlation_matrix.columns:
    # Сортуємо кореляції
    menu_correlations = correlation_matrix[menu_column].sort_values(ascending=Fa

    # Виберемо 10 найбільших кореляцій
    top_10_correlations = menu_correlations[1:11]
    print("Топ-10 кореляцій для стовпця '39 Menu Eatgirvid':")
    print(top_10_correlations)
else:
    print(f"Стовпець '{menu_column}' не знайдено в матриці кореляцій.")
```

```
Топ-10 кореляцій для стовпця '39 Menu Eatgirvid':
Menu Eatgiroil                          0.583221
Menu Termin Vagitnosti                  0.254927
Number Covidpostsindromtimeweeks        0.221860
Checkbox Eatgirtvar                     0.209363
83 misec Number Covidpostsindrommons    0.105967
17 Menu Education                       0.102353
99 Radio Addanketanew                   0.092214
29 Radio Eatovoch                       0.089921
  Status                                0.084659
53 Radio Tiskizmer                      0.077179
Name: 39 Menu Eatgirvid, dtype: float64
```

In [45]:
```python
#Перевірка на мультиколінеарність

from statsmodels.stats.outliers_influence import variance_inflation_factor
import pandas as pd
import numpy as np

df_cleaned = df_cleaned.fillna(0)  # Заповнення NaN
df_cleaned = df_cleaned.replace([np.inf, -np.inf], 0)  # Заміна нескінченностей

# Розраховуємо VIF для кожної змінної
vif_data = pd.DataFrame()
vif_data['Feature'] = df_cleaned.columns
vif_data['VIF'] = [
    variance_inflation_factor(df_cleaned.values, i)
    for i in range(df_cleaned.shape[1])
]

print("Значення VIF для кожної змінної:")
print(vif_data.sort_values(by='VIF', ascending=False))
```

```
Значення VIF для кожної змінної:
                  Feature          VIF
1                  2 Date  1030.297053
138  85 Radio Covidvakcina    47.399213
6        6 Radio Vagitnist    40.999733
4             4 Radio Stat    38.120245
10          8A Date Vidizd    31.106891
..                     ...          ...
113    61 Menu Poglmisgrud     1.210884
127      75 Menu Bilzakrep     1.210038
101      54 Menu Pulsizmer     1.197315
51        26 Menu Fatwater     1.187583
65       34 Menu Eatsoltwo     1.178033

[164 rows x 2 columns]
```

In [47]:

```python
# КРОК 1 тут присутня мультиколінеарність, необхідно зменшти її
#Автоматичне видалення змінних із високим VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor
import pandas as pd

# Функція для обчислення VIF
def calculate_vif(df):
    vif_data = pd.DataFrame()
    vif_data['Feature'] = df.columns
    vif_data['VIF'] = [variance_inflation_factor(df.values, i) for i in range(df
    return vif_data

# Функція для автоматичного видалення змінних із високим VIF
def reduce_multicollinearity(df, threshold=50):
    vif_data = calculate_vif(df)
    while vif_data['VIF'].max() > threshold:
        # Знайдемо змінну із максимальним VIF
        max_vif_feature = vif_data.sort_values('VIF', ascending=False).iloc[0]
        print(f"Видаляємо змінну: {max_vif_feature['Feature']} с VIF = {max_vif_

        # Видаляємо змінну із максимальним VIF
        df = df.drop(columns=[max_vif_feature['Feature']])

        # Перераховуємо VIF
        vif_data = calculate_vif(df)

    return df

# Застосовуємо функцію до нових даних
df_reduced = reduce_multicollinearity(df_cleaned)

# Виводимо оновлену матрицю VIF
vif_data = calculate_vif(df_reduced)
print("Матрица VIF после снижения мультиколлинеарности:")
print(vif_data.sort_values(by='VIF', ascending=False))
```

```
Видаляємо змінну: 2 Date с VIF = 1030.297053161064
Видаляємо змінну: _ Status с VIF = 184.6328537578929
Видаляємо змінну: 85 Radio Covidvakcina с VIF = 152.77511708192057
Видаляємо змінну: 6 Radio Vagitnist с VIF = 132.3990017911976
Видаляємо змінну: 44 dlina Number Dovgtila с VIF = 102.14406460579887
Видаляємо змінну: 44 massa Number Masatila с VIF = 99.94365736346526
Видаляємо змінну: 76 Radio Covid с VIF = 63.91132960405603
Видаляємо змінну: 8 Radio Misceprogiv с VIF = 50.34267214118852
Матрица VIF после снижения мультиколлинеарности:
                              Feature        VIF
149                  94 Radio Fizkult  49.428104
126        82 Radio Covidpostsindrom  40.442025
97           55A Radio Cukorizmerbig  39.748142
66                 39 Menu Eatgirvid  36.030514
52                 29 Radio Eatovoch  31.026065
..                               ...        ...
68               Checkbox Eatgirtvar   1.430708
1                      3 Date Howold   1.382899
78      45 zmensh Number Vagachengzm   1.351106
4             Menu Termin Vagitnosti   1.237458
36   21 EXTRA Textarea Badparenttext   1.226081

[156 rows x 2 columns]
```

In [49]:

```python
# Факторний аналіз із необхідними тестами
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.decomposition import FactorAnalysis
from factor_analyzer import FactorAnalyzer
from scipy.stats import bartlett


# 1. Проверка пригодности данных для факторного анализа
# KMO тест
from factor_analyzer.factor_analyzer import calculate_kmo
kmo_all, kmo_model = calculate_kmo(df_cleaned)
print(f"KMO тест: {kmo_model}")  # Если значение близко к 1, то данные пригодны

# Bartlett's тест
chi_square_value, p_value = bartlett(*[df_cleaned[col] for col in df_cleaned.col
print(f"Bartlett's test: p-value = {p_value}")  # Если p-value < 0.05, то данные
```

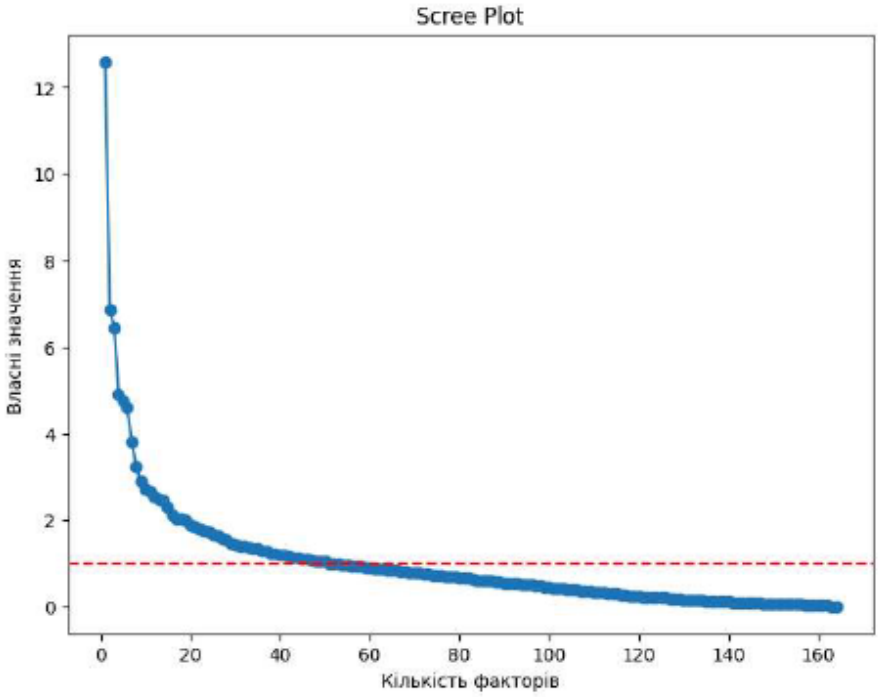
```
KMO тест: 0.7810240370201796
Bartlett's test: p-value = 0.0
```

In [51]:

```python
# 2. Визначення оптимальної кількості факторів за допомогою метода Kaiser и Scre

# Метод Kaiser
fa = FactorAnalyzer()
fa.fit(df_cleaned)
eigenvalues, _ = fa.get_eigenvalues()  # Получаем только собственные значения

print("Власні значення факторів:", eigenvalues)

# Вибір факторів, для яких власні значення > 1
n_factors_kaiser = sum(eigenvalues > 1)
print(f"Оптимальна кількість факторів по методу Kaiser: {n_factors_kaiser}")
```

```python
# Scree Plot
plt.figure(figsize=(8, 6))
plt.plot(range(1, len(eigenvalues) + 1), eigenvalues, marker='o')
plt.axhline(y=1, color='r', linestyle='--')
plt.title('Scree Plot')
plt.xlabel('Кількість факторів')
plt.ylabel('Власні значення')
plt.show()
```

```
Власні значення факторів: [1.25909280e+01 6.85300010e+00 6.45696701e+00 4.90721
016e+00
 4.77128176e+00 4.60815134e+00 3.81688884e+00 3.24771830e+00
 3.09872181e+00 2.71791433e+00 2.67979926e+00 2.56204983e+00
 2.48008325e+00 2.44481797e+00 2.30563425e+00 2.13601786e+00
 2.04051776e+00 2.02691919e+00 2.01056231e+00 1.88856285e+00
 1.85342882e+00 1.79576117e+00 1.77941024e+00 1.72852787e+00
 1.69095122e+00 1.65864691e+00 1.59588659e+00 1.56643081e+00
 1.47032436e+00 1.45267670e+00 1.40624611e+00 1.40136587e+00
 1.37503310e+00 1.35969747e+00 1.35483611e+00 1.29866159e+00
 1.28309195e+00 1.24545921e+00 1.24313005e+00 1.21607826e+00
 1.19471373e+00 1.15894651e+00 1.14623356e+00 1.13016018e+00
 1.12138316e+00 1.09872570e+00 1.07315808e+00 1.06331180e+00
 1.04484915e+00 1.04194233e+00 1.00407293e+00 9.95575612e-01
 9.82888269e-01 9.70099137e-01 9.55873477e-01 9.32593039e-01
 9.27139061e-01 9.18495264e-01 9.09491945e-01 8.92226147e-01
 8.83173632e-01 8.77793428e-01 8.71050084e-01 8.56308597e-01
 8.46093193e-01 8.31773460e-01 8.20776570e-01 8.02689561e-01
 7.84800339e-01 7.73922043e-01 7.69416589e-01 7.49503781e-01
 7.41327996e-01 7.34487737e-01 7.26738285e-01 7.19686726e-01
 7.00728488e-01 6.86463622e-01 6.78986602e-01 6.72978960e-01
 6.64025557e-01 6.54024375e-01 6.27586804e-01 6.14236532e-01
 6.05399708e-01 6.00431845e-01 5.91063901e-01 5.77906311e-01
 5.65929890e-01 5.56700515e-01 5.49227066e-01 5.41634535e-01
 5.30207729e-01 5.16810939e-01 5.12569804e-01 5.04169618e-01
 4.88995684e-01 4.81362372e-01 4.63484633e-01 4.51453378e-01
 4.37845766e-01 4.30028905e-01 4.20945987e-01 4.07723225e-01
 4.04669413e-01 3.79061661e-01 3.77406081e-01 3.67554258e-01
 3.55187725e-01 3.40636126e-01 3.37403968e-01 3.30400105e-01
 3.14861949e-01 3.07696824e-01 2.93779065e-01 2.76472982e-01
 2.69913007e-01 2.57823546e-01 2.43164273e-01 2.35806819e-01
 2.26772058e-01 2.21353515e-01 2.17690239e-01 2.14066188e-01
 2.03327764e-01 2.02855643e-01 1.92511427e-01 1.83795186e-01
 1.83092623e-01 1.66786201e-01 1.61018885e-01 1.58804363e-01
 1.50889515e-01 1.45260876e-01 1.38319153e-01 1.32598983e-01
 1.28532734e-01 1.21584938e-01 1.15422689e-01 1.09653476e-01
 1.04011001e-01 1.03140579e-01 9.67857533e-02 9.43377197e-02
 9.14598873e-02 8.52582915e-02 8.02834747e-02 7.67472113e-02
 7.43124527e-02 6.71441419e-02 6.30211450e-02 5.95622543e-02
 5.82420865e-02 5.65495548e-02 5.44723707e-02 5.02669616e-02
 4.58085097e-02 3.38794293e-02 3.12316367e-02 2.86953675e-02
 2.79602379e-02 2.24670126e-02 1.31329979e-02 1.21306065e-02]
Оптимальна кількість факторів по методу Kaiser: 51
```



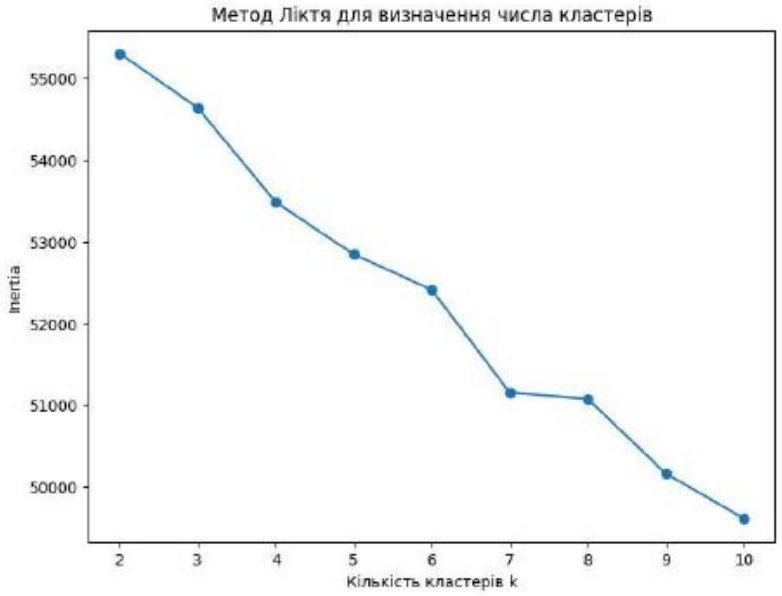
In [52]:
```python
# 4. Факторний аналіз для зниження розмірності

def apply_factor_analysis(df, n_components=10):
    fa = FactorAnalysis(n_components=n_components)
    reduced_data = fa.fit_transform(df)  # Перетворюємо дані в нові компоненти
    reduced_df = pd.DataFrame(reduced_data, columns=[f'Factor{i+1}' for i in ran
    return reduced_df

df_factors = apply_factor_analysis(df_cleaned, n_components=n_factors_kaiser)

print("Данные после факторного анализа:")
print(df_factors.head())
```

```
Данные после факторного анализа:
    Factor1   Factor2   Factor3   Factor4   Factor5   Factor6   Factor7  \
0 -0.875909  0.191411 -1.583604 -0.367624  0.049027 -0.310774  0.966282
1  1.143784  0.013946 -1.460653  0.482180  0.675690 -0.407117  0.933653
2 -0.875909 -0.149623 -1.615122  1.125848 -0.016529 -0.318817  0.866685
3 -0.875909 -0.559561 -1.653665 -0.651995  0.087766 -0.367413  0.750749
4 -0.875909 -0.054793 -1.604947 -0.630320  0.051980  2.190429  0.230557

    Factor8   Factor9  Factor10  ...  Factor42  Factor43  Factor44  Factor45  \
0 -0.781864 -0.326432  1.050925  ...  0.089993  0.268490  2.060011 -0.330116
1 -0.303808  0.316646  1.130155  ... -0.964817  0.300080  2.257315  0.356357
2 -1.427745  0.941214 -0.568585  ... -0.321819  0.222420  1.290260  0.244741
3 -0.590738  0.364738  1.162981  ... -0.728293 -2.700252 -0.067598  1.777551
4 -0.557870 -0.086354  2.531623  ...  0.322252 -1.699326 -0.813449  2.016455

   Factor46  Factor47  Factor48  Factor49  Factor50  Factor51
0  0.458316 -0.501626 -1.052536  0.042925 -0.376196 -0.328199
1  1.690038  1.057112 -2.144438 -0.239127 -0.027710  0.141331
2 -0.818950 -0.884792 -0.391015  1.446907 -0.074760 -0.928879
3 -0.363336 -0.591514 -0.524973  1.006552  0.168581  0.620474
4 -0.210368  0.228663 -0.656197  1.142891 -0.194321  0.039486

[5 rows x 51 columns]
```

In [53]:
```
#Кластеризація
```

In [56]:
```
#Нормалізація
from sklearn.preprocessing import StandardScaler

# Нормалізація факторів
scaler = StandardScaler()
df_factors_scaled = scaler.fit_transform(df_factors)
```

In [57]:
```
#Кластеризація даних
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

inertia = []
range_k = range(2, 11)  # Від 2 до 10 кластерів

for k in range_k:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df_factors_scaled)
    inertia.append(kmeans.inertia_)

# Візуалізація методу ліктя
plt.figure(figsize=(8, 6))
plt.plot(range_k, inertia, marker='o')
plt.title('Метод Ліктя для визначення числа кластерів')
plt.xlabel('Кількість кластерів k')
plt.ylabel('Inertia')
plt.show()
```



In [68]:
```
optimal_k = 5 #  знайдене значення k
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
df_factors['Cluster'] = kmeans.fit_predict(df_factors_scaled)

print("Кластери успішно додано до даних")
```

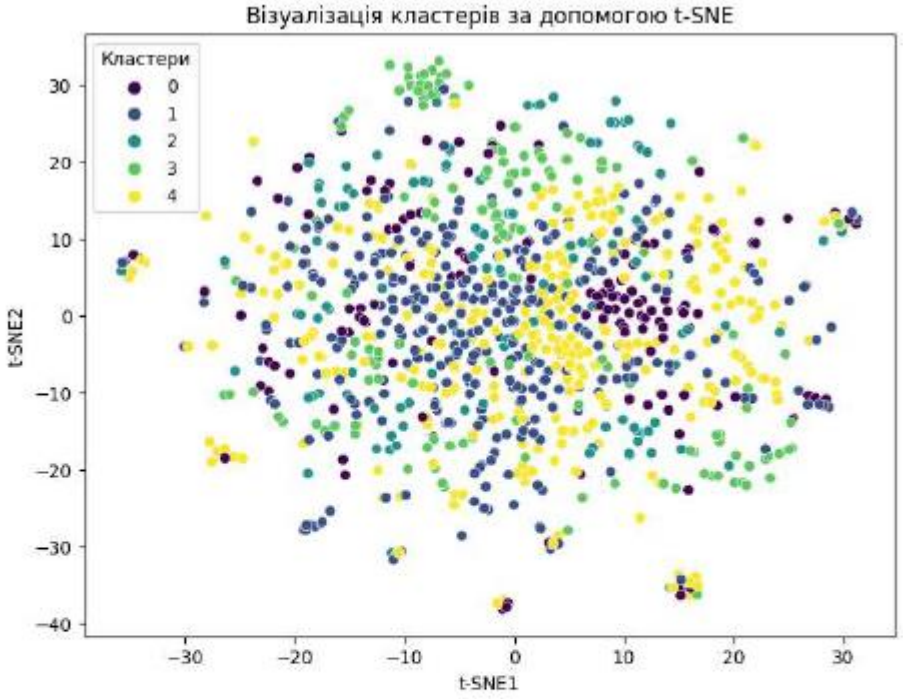
```
Кластери успішно додано до даних
```

In [69]:
```
#Аналіз кластерів
cluster_summary = df_factors.groupby('Cluster').mean()
print(cluster_summary)
```

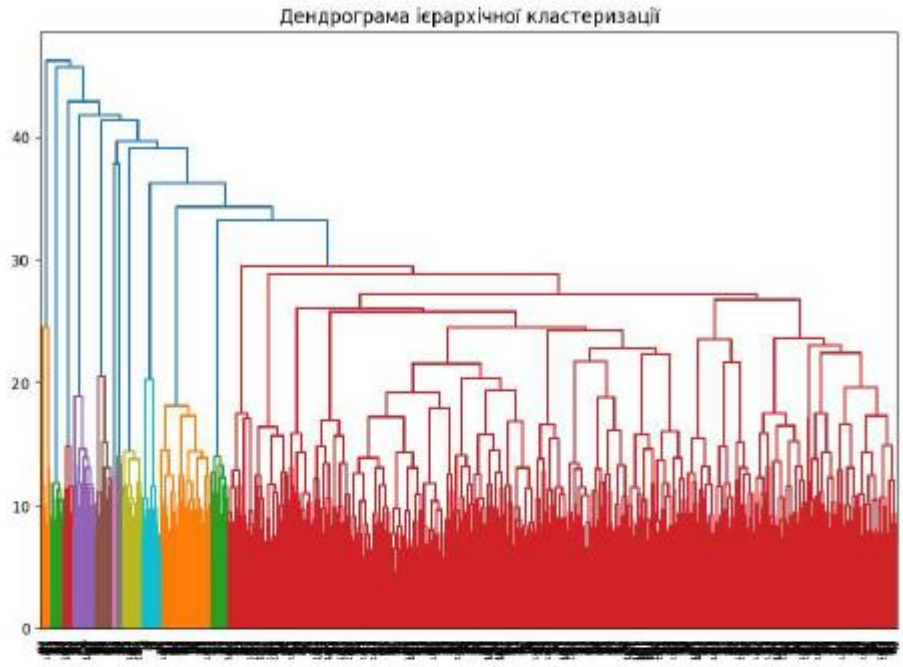
```
          Factor1   Factor2   Factor3   Factor4   Factor5   Factor6   Factor7
\
Cluster
0        0.209709  0.117162 -0.026657 -0.112527 -0.014872  2.261624 -0.456537
1       -0.020583 -0.046135  0.089744 -0.045212 -0.017207 -0.436832  0.078338
2       -0.027638 -0.128739  0.016542 -0.025438  0.010480  0.152058  0.209678
3       -0.056191  0.052230  0.199609  0.092108 -0.021632 -0.369274 -0.085026
4       -0.032391  0.016094 -0.193012  0.058916  0.032220 -0.411869  0.088950

          Factor8   Factor9  Factor10  ...  Factor44  Factor45  Factor46  \
Cluster                                ...
0       -0.168461  0.115127 -0.041829  ... -0.002971  0.019800  0.055016
1       -0.197752  0.160576  0.202238  ... -0.115659 -0.258542  0.143761
2        0.075968 -0.659846  0.348356  ... -0.056042  0.010202 -0.017134
3        0.074932  0.071463 -0.099076  ...  0.073912 -0.004911 -0.250710
4        0.216244 -0.016115 -0.269882  ...  0.104892  0.262066 -0.038412

          Factor47  Factor48  Factor49  Factor50  Factor51    t-SNE1    t-SNE2
Cluster
0       -0.061475  0.014867  0.014119  0.027728  0.015303  2.962193  0.100873
1        0.125325  0.128214  0.352581  0.093453  0.282723 -2.545269 -2.250581
2        0.023436  0.004582  0.014869  0.027614  0.026419 -1.079095  3.033828
3        0.018331 -0.121752 -0.104343 -0.009404 -0.179142  0.256919  3.958704
4       -0.122623 -0.079546 -0.328188 -0.115905 -0.220308  1.172422 -2.657929

[5 rows x 53 columns]
```

In [70]:

```python
from sklearn.manifold import TSNE
import seaborn as sns

# Зменшення вимірності до 2 компонентів
tsne = TSNE(n_components=2, random_state=42)
tsne_result = tsne.fit_transform(df_factors_scaled)

# Додаємо t-SNE координати
df_factors['t-SNE1'] = tsne_result[:, 0]
df_factors['t-SNE2'] = tsne_result[:, 1]

# Графік кластерів
plt.figure(figsize=(8, 6))
sns.scatterplot(x='t-SNE1', y='t-SNE2', hue='Cluster', data=df_factors, palette=
plt.title('Візуалізація кластерів за допомогою t-SNE')
plt.xlabel('t-SNE1')
plt.ylabel('t-SNE2')
plt.legend(title='Кластери')
plt.show()
```



In [61]:
```
#Ієрархічна кластеризація
from scipy.cluster.hierarchy import dendrogram, linkage

# Побудова дендрограми
linked = linkage(df_factors_scaled, method='ward')
plt.figure(figsize=(10, 7))
dendrogram(linked)
plt.title('Дендрограма ієрархічної кластеризації')
plt.show()
```



In [ ]:

In [71]:
```
#Аналіз конкретно значень 2 і 4
# Створимо 2 підмножини, бінарізація даних
df_reduced['is_2'] = (df_reduced['39 Menu Eatgirvid'] == 2).astype(int)
df_reduced['is_4'] = (df_reduced['39 Menu Eatgirvid'] == 4).astype(int)
```

In [74]:
```
#Знаходимо кореляції для значень
# Обчислення кореляцій для значення 2
correlations_2 = df_reduced.corr()['is_2']

# Топ-5 змінних, що найбільше корелюють із значенням 2
top_10_correlations_2 = correlations_2.abs().sort_values(ascending=False).head(1
print("Топ-10 кореляцій із значенням 2:")
print(top_10_correlations_2)


# Обчислення кореляцій для значення 4
correlations_4 = df_reduced.corr()['is_4']

# Топ-5 змінних, що найбільше корелюють із значенням 4
top_10_correlations_4 = correlations_4.abs().sort_values(ascending=False).head(1
print("Топ-10 кореляцій із значенням 4:")
print(top_10_correlations_4)
```

```
Топ-10 кореляцій із значенням 2:
is_2                      1.000000
39A Number Eatgirmaslo    0.938977
is_4                      0.718587
39 Menu Eatgirvid         0.663260
Menu Eatgiroil            0.540721
16 Menu Havejob           0.096771
47 Menu Sonhowmach        0.095652
27 Menu Eatfruct          0.091105
35 Menu Eatsol            0.089906
8A Date Vidizd            0.084435
Name: is_2, dtype: float64
Топ-10 кореляцій із значенням 4:
is_4                      1.000000
39 Menu Eatgirvid         0.786277
Menu Eatgiroil            0.744645
is_2                      0.718587
39A Number Eatgirmaslo    0.673183
Checkbox Eatgirtvar       0.224869
16 Menu Havejob           0.105819
30 Number Eatovochnumb    0.097000
4 Radio Stat              0.092692
53A Menu Tiskizmerpokaz   0.089851
Name: is_4, dtype: float64
```

In [15]:
```python
# Перевіримо, які значення містить в собі стовпець '39 Menu Eatgirvid'
if '39 Menu Eatgirvid' in df_cleaned.columns:  # Перевіряємо наявність стовпця у
    print("Унікальні значення в стовпці '39 Menu Eatgirvid':")
    print(df_cleaned['39 Menu Eatgirvid'].unique())  # Виводимо унікальні значен
else:
    print("Стовпець '39 Menu Eatgirvid' відсутній в DataFrame.")
```

```
Унікальні значення в стовпці '39 Menu Eatgirvid':
[4 2 5 1 3 0]
```

In [46]:
```python
#Факторний аналіз

# Вибірка для значень 2 і 4
filtered_data = df_cleaned[df_cleaned['39 Menu Eatgirvid'].isin([2, 4])]

from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer

# Вибір числових стовпців
numeric_columns = filtered_data.select_dtypes(include=['float64', 'int64']).colu
numeric_data = filtered_data[numeric_columns]
imputer = SimpleImputer(strategy='mean')
numeric_data = imputer.fit_transform(numeric_data)
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_data)


#Тест KMO і Тест Бартлетта
from factor_analyzer import calculate_kmo
from scipy.stats import bartlett

kmo_all, kmo_model = calculate_kmo(scaled_data)
print(f"KMO (адекватність факторного аналізу): {kmo_model}")
```

```
chi_square_value, p_value = bartlett(*scaled_data.T)
print(f"Bartlett's test: p-value = {p_value}")
```

```
KMO (адекватність факторного аналізу): 0.5751287530833329
Bartlett's test: p-value = 1.0
```

In [ ]: