Лабораторна робота № 1. Отримання навичок роботи в середовищі Python

Недашківська Н.І.

1 Варіанти завдань

Увага ! При написанні коду використовувати, там де це доречно, спеціалізовані функції бібліотеки NumPy, універсальні функції, функції транслювання (broadcasting).

1. Дано масив T, який складається з N рядків, які відповідають прикладам, і m стовпчиків, які відповідають ознакам. Відомо, що ознака x_h приймає значення з множини $\{c_{h1}, c_{h2}, ..., c_{hq_h}\}$. Дано вектор y розмірності N, елементи якого приймають значення з множини $S = \{s_1, s_2, ..., s_v\}$ (мітки класів для прикладів). Знайти ознаку x_h^*, для якої наступний вираз приймає мінімальне значення:

$$G(x_h) = \sum_{i=1}^{q_h} \frac{|T_i|}{N} H(T_i, S),$$

де T_i - підмножина прикладів, для яких ознака x_h приймає значення c_{hi}, $|A|$ - потужність множини A, $H(A, S)$ - ентропія множини A по відношенню до властивості S:

$$H(A, S) = -\sum_{i=1}^{v} \frac{k_i}{|A|} \log_2 \frac{k_i}{|A|},$$

де властивість S може приймати v різних значень, кожне з яких - в k_i випадках.

2. Дано масив T, який складається з N рядків, які відповідають прикладам, і m стовпчиків, які відповідають ознакам. Відомо, що ознака x_h приймає значення з множини $\{c_{h1}, c_{h2}, ..., c_{hq_h}\}$. Дано вектор y розмірності N, елементи якого приймають значення з множини $S = \{s_1, s_2, ..., s_v\}$ (мітки класів для прикладів). Знайти ознаку x_h^*, для якої наступний вираз приймає мінімальне значення:

$$G(x_h) = \sum_{i=1}^{q_h} \frac{|T_i|}{N} H(T_i, S),$$

де T_i - підмножина прикладів, для яких ознака x_h приймає значення c_{hi}, $|A|$ - потужність множини A, $H(A,S)$ - індекс Джині множини A по відношенню до властивості S:

$$H(A,S) = 1 - \sum_{i=1}^{v} \left(\frac{k_i}{|A|}\right)^2,$$

де властивість S може приймати v різних значень, кожне з яких - в k_i випадках.

3. Дано масив T, який складається з N рядків, які відповідають прикладам, і m стовпчиків, які відповідають ознакам. Відомо, що ознака x_h приймає значення з множини $\{c_{h1}, c_{h2}, ..., c_{hq_h}\}$. Дано вектор y розмірності N, елементи якого приймають значення з множини $S = \{s_1, s_2, ..., s_v\}$ (мітки класів для прикладів). Знайти ознаку x_h^* та значення цієї ознаки c_{hi}^*:

$$\mathrm{c}_{hi}^* = \arg\max_{h,i} \frac{p_2(y = s_j | x_h = c_{hi})}{p_1(x_h = c_{hi})},$$

де s_j - задано, $p_1(x_h = c_{hi})$ - кількість прикладів, для яких ознака x_h приймає значення c_{hi}, $p_2(y = s_j | x_h = c_{hi})$ - кількість прикладів, які належать класу s_j і ознака x_h приймає значення c_{hi}.

4. Дано масив T, який складається з N рядків, які відповідають прикладам, і m стовпчиків, які відповідають ознакам. Відомо, що ознака x_h приймає значення з множини $\{c_{h1}, c_{h2}, ..., c_{hq_h}\}$. Дано вектор y розмірності N, елементи якого приймають значення з множини $S = \{s_1, s_2, ..., s_v\}$ (мітки класів для прикладів). Знайти ознаку x_h^* та значення цієї ознаки c_{hi}^*:

$$\mathrm{c}_{hi}^* = \arg\min_{h,i} Er(h,i),$$

$$Er(h,i) = \frac{p_3(y \neq s_j^* | x_h = c_{hi})}{p_1(x_h = c_{hi})},$$

$$s_j^* = \arg\max_{j} p_2(y = s_j | x_h = c_{hi}),$$

де $p_1(x_h = c_{hi})$ - кількість прикладів, для яких ознака x_h приймає значення c_{hi}, $p_2(y = s_j | x_h = c_{hi})$ - кількість прикладів, які належать класу s_j і ознака x_h приймає значення c_{hi}, s_j^* - найбільш імовірний клас за умови що ознака x_h приймає значення c_{hi}.

5. Дано масив T, який складається з N рядків, які відповідають прикладам, і m стовпчиків, які відповідають ознакам. Відомо, що ознака x_h приймає значення $\{c_{h1}, c_{h2}, ..., c_{hq_h}\}$. Дано вектор y розмірності N, елементи якого приймають значення з множини $S = \{s_1, s_2, ..., s_v\}$ (мітки класів для прикладів). Знайти значення s_k^* (найбільш імовірний клас) для нового

прикладу, який характеризується заданими значеннями ознак $x_1 = a_1$, $x_2 = a_2$, ..., $x_m = a_m$:

$$s_k^* = \arg\max_{s_k \in S} p(y = s_k) \prod_{i=1}^{N} p(x_i = a_i | y = s_k),$$

де a_i - задані, $p(y = s_k)$ - кількість прикладів, які належать класу s_k, $p(x_i = a_i | y = s_k)$ - кількість прикладів, у яких ознака x_i приймає значення a_i, серед тих, що належать класу s_k.

Захист: Знайти ознаку x_h^* та значення цієї ознаки c_{hi}^*:

$$\mathrm{c}_{hi}^* = \arg\max_{h,i} \frac{p_2(y = s_j | x_h = c_{hi})}{p_1(x_h = c_{hi})},$$

де s_j - задано, $p_1(x_h = c_{hi})$ - кількість прикладів, для яких ознака x_h приймає значення c_{hi}, $p_2(y = s_j | x_h = c_{hi})$ - кількість прикладів, які належать класу s_j і ознака x_h приймає значення c_{hi}.

6. Дано масив $T = \{(t_i) | t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$, $x_{ik} \in R$, де приклад t_i характеризується m ознаками. Для цих даних розрахувати матриці відстаней: евклідової D_2, хемінга D_H, чебишева D_∞, пікову D_P та махаланобіса D_M:

$$D_2(t_p, t_q) = \sqrt{\sum_{k=1}^{m} (x_{pk} - x_{qk})^2}$$

$$D_H(t_p, t_q) = \sum_{k=1}^{m} |x_{pk} - x_{qk}|$$

$$D_\infty(t_p, t_q) = \max_{k=1,...,m} |x_{pk} - x_{qk}|$$

$$D_P(t_p, t_q) = \frac{1}{m} \sum_{k=1}^{m} \frac{|x_{pk} - x_{qk}|}{x_{pk} + x_{qk}}$$

$$D_M(t_p, t_q) = \sqrt{(t_p - t_q)^T S^{-1} (t_p - t_q)},$$

де S - матриця коваріації.

7. Дано масив $T = \{(t_i) | t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$, $x_{ik} \in R$, де приклад t_i характеризується m ознаками. Об'єднати приклади в кластери за наступним алгоритмом:

1) $C := T$, множина кластерів C співпадає з початковою множиною прикладів,

2) Поки в C більше одного елементу:

- вибираємо два кластери $c_p, c_q \in C$, відстань між якими мінімальна,
- об'єднуємо c_p і c_q у новий кластер c_{pq}, змінюємо C за правилом:

$$C := C \cup c_{pq} \setminus \{c_p, c_q\},$$

Відстань між кластерами:

$$d_{rs} = \frac{d_{ps} + d_{qs}}{2},$$

де d_{rs} - відстань від нового кластера c_r, який утворено об'єднанням c_p і c_q, до іншого кластера c_s.

Надрукувати множину кластерів C і матрицю відстаней між отриманими кластерами.

8. Розглянути умову попередньої задачі. Надрукувати множину кластерів C і матрицю відстаней між отриманими кластерами, якщо відстань між кластерами розраховується за формулою:

$$d_{rs} = \frac{d_{ps} + d_{qs}}{2} - \frac{|d_{ps} - d_{qs}|}{2},$$

де d_{rs} - відстань від нового кластера c_r, який утворено об'єднанням c_p і c_q, до іншого кластера c_s.

9. Дано масив $T = \{(t_i)|t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$, $x_{ij} \in R$, де приклад t_i характеризується m ознаками. Задано кількість кластерів $2 \leq g \leq N$. Розрахувати центри кластерів за формулою (класичний метод к-середніх):

$$c_k = \frac{\sum_{i=1}^{N} u_{ki} t_i}{\sum_{i=1}^{N} u_{ki}}, k = 1, ..., g,$$

де $U = \{(u_{ki})|k = 1, ..., g, i = 1, ..., N\}$ - випадковим чином задана матриця початкового розбиття, $u_{ki} \in \{0, 1\}$, $\sum_{k=1}^{g} u_{ki} = 1$, $\sum_{i=1}^{N} u_{ki} < N$.

Перерахувати матрицю розбиття:

$u_{ki} = 1$ якщо $d(t_i, c_k) = \min_{l=1,...,g} d(t_i, c_l)$,

$u_{ki} = 0$ в іншому випадку,

за умови, що $d(t_i, c_k)$ - евклідова відстань між векторами.

Виконати декілька ітерацій з уточнення центрів кластерів.

10. Задано неорієнтовний граф G з V вершинами, де ваги дуг d_{ij} відомі для $\forall i, j = 1, ..., V$, $i \neq j$ і позначають відстані між об'єктами. Задано поріг близькості $\sigma \in [\min d_{ij}, \max d_{ij}]$. Знайти множину кластерів на основі графу G, використовуючи алгоритм:

1) Вилучити з графа ребра, ваги яких перевищують заданий поріг близькості σ.

2) Компонента зв'язності графу – підмножина вершин графу, в якій будь-які вершини можна поєднати шляхом, який цілком належить цій підмножині.

Знайти компоненти зв'язності отриманого графа, вони і будуть шуканими кластерами.

11. Задано неорієнтовний граф G з V вершинами, де ваги дуг d_{ij} відомі для $\forall i, j = 1, ..., V$, $i \neq j$. Побудувати мінімальне покриваюче дерево - підграф J графу G, використовуючи алгоритм Крускала:

    1) Відсортувати ребра в порядку зростання їх ваг. $J := \varnothing$.

    2) Додавати по одному ребру до J, якщо це ребро не утворює цикл з наявними ребрами.

    Пояснення: першим додається ребро мінімальної ваги, далі наступне із списку відсортованих ребер і т.д. Якщо деяке ребро утворює цикл з наявними ребрами, то воно пропускається (не додається до J) і здійснюється перехід до наступного ребра в списку відсортованих ребер.

    3) Виконувати крок 2 до тих пір поки до J не буде додано $V - 1$ ребро.

12. Задано неорієнтовний граф G з V вершинами, де ваги дуг d_{ij} відомі для $\forall i, j = 1, ..., V$, $i \neq j$. Побудувати мінімальне покриваюче дерево - підграф J графу G, використовуючи алгоритм Прима:

    1) Вибрати будь-яку вершину графу G і додати її до J.

    2) Додати до J ребро з найменшою вагою, яке з'єднує вершину підграфу J з вершиною, яка не належить J.

    3) Виконувати крок 2 до тих пір поки до J не буде додано $V - 1$ ребро.

13. Розглянути критерій якості кластеризації - коефіцієнт розбиття:

$$PC = \frac{\sum_{j=1}^{N} \sum_{k=1}^{g} u_{kj}^2}{N}.$$

    Задати: N - кількість об'єктів, які кластеризуються, $1 \leq g \leq N$ - кількість кластерів, $U = \{(u_{kj}) | k = 1, ..., g, j = 1, ..., N\}$ - матриця розбиття на кластери (використовується у нечіткому методі к-середніх), $u_{kj} \in [0, 1]$ - це ступінь належності j-го об'єкту k-му кластеру, причому $u_{kj} = 1$ означає повну приналежність, $u_{kj} = 0.5$ означає приналежність до k-го кластеру зі ступенем 0.5, $\sum_{k=1}^{g} u_{kj} = 1$, $\sum_{j=1}^{N} u_{kj} < N$.

    Використовуючи результати моделювання великої кількості матриць розбиття, показати, що

$$PC \in \left[\frac{1}{g}, 1\right].$$

14. Розглянути критерій якості кластеризації - ентропію розбиття:

$$PE = -\frac{\sum_{j=1}^{N}\sum_{k=1}^{g} u_{kj}\ln u_{kj}}{N}.$$

Задати: N - кількість об'єктів, які кластеризуються, $1 \le g \le N$ - задана кількість кластерів, $U = \{(u_{kj})|k = 1, ..., g, j = 1, ..., N\}$ - матриця розбиття на кластери (використовується, наприклад, у нечіткому методі к-середніх), $u_{kj} \in [0, 1]$ - це ступінь належності j-го об'єкту k-му кластеру, причому $u_{kj} = 1$ означає повну приналежність, $u_{kj} = 0.5$ означає приналежність до k-го кластеру зі ступенем 0.5, $\sum_{k=1}^{g} u_{kj} = 1$, $\sum_{j=1}^{N} u_{kj} < N$.

Використовуючи результати моделювання великої кількості матриць розбиття, показати, що

$$PE \in [0, \ln g].$$

15. Згенерувати N точок в R^2 так, щоб вони утворювали віддалені один від одного скупчення, $1 \le g^* \le N$ - задана кількість кластерів. Для цих точок згенерувати $U^* = \{(u_{kj})|k = 1, ..., g^*, j = 1, ..., N\}$ - матрицю розбиття точок на кластери (використовується, наприклад, у нечіткому методі к-середніх), $u_{kj} \in [0, 1]$ - це ступінь належності j-ї точки k-му кластеру, причому $u_{kj} = 1$ означає повну приналежність, $u_{kj} = 0.5$ означає приналежність до k-го кластеру зі ступенем 0.5, $\sum_{k=1}^{g^*} u_{kj} = 1$, $\sum_{j=1}^{N} u_{kj} < N$.

Розглянути декілька результатів кластеризації точок, які задаються матрицями розбиття:

- еталонна кластеризація, яка задається U^* і відповідає початковим правилам генерування точок,

- зашумлені кластеризації, в яких окремі точки віднесені до інших кластерів. Розглянути також випадки коли кількість кластерів g не співпадає з початково згенерованою g^*.

Показати, що на найкращому розбитті U^* індекс чіткості CI приймає найбільше значення:

$$CI = \frac{gPC - 1}{g - 1},$$

$$PC = \frac{\sum_{j=1}^{N}\sum_{k=1}^{g} u_{kj}^2}{N}.$$

16. Розглянути умову попереднього варіанту. Дослідити, яке значення приймає модифікована ентропія розбиття PE_M на найкращому розбитті U^* :

$$PE_M = \frac{PE}{\ln g},$$

$$PE = -\frac{\sum_{j=1}^{N}\sum_{k=1}^{g} u_{kj}\ln u_{kj}}{N}.$$

17. Розрахувати індекс ефективності кластеризації:

$$PI = \sum_{j=1}^{N} \sum_{k=1}^{g} u_{kj}^2 (d^2(\bar{t}, c_k) - d^2(t_j, c_k)),$$

- $T = \{(t_i)|t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$ - множина об'єктів, які кластеризуються, $x_{ik} \in R$,
- $\bar{t}$ - вибіркове середнє об'єктів $t_i \in T$,
- $2 \leq g \leq N$ - задана кількість кластерів,
- $U = \{(u_{kj})|k = 1, ..., g, j = 1, ..., N\}$ - задана матриця розбиття згідно з класичним методом к-середніх, яка задовольняє умовам $u_{kj} \in \{0, 1\}$, причому $u_{kj} = 1$ означає приналежність j-го об'єкту k-му кластеру, $\sum_{k=1}^{g} u_{kj} = 1$, $\sum_{j=1}^{N} u_{kj} < N$,
- $\{c_k|k = 1, ..., g\}$ - задані центри кластерів,
- $d^2(t_j, c_k)$ - квадрат евклідової відстані між векторами.

18. Дано масив $T = \{(t_i)|t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$ об'єктів, які потрібно кластеризувати, $x_{ik} \in R$. Задано параметр $\rho > 0$. В якості міри близькості вибрано евклідову відстань $d(t_i, t_j)$. Знайти множину кластерів за наступними етапами (алгоритм Форел):

1) Ініціалізувати множину некластеризованих точок $U := T$.

2) Поки є некластеризовані точки, тобто $U \neq \varnothing$:

- випадковим чином вибрати $t_0 \in U$,
- повторювати:
  - утворити кластер – сферу з центром t_0 і радіусом ρ:

$$C_0 := \{t_i \in T|d(t_i, t_0) \leq \rho\},$$

  - помістити центр сфери в центр мас кластера:

$$t_0 := \frac{1}{|C_0|} \sum_{t_i \in C_0} t_i,$$

- поки центр t_0 не стабілізується,
- відмітити всі точки множини C_0 як кластеризовані: $U := U \setminus C_0$.

19. Дано масив $T = \{(t_i)|t_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., N\}$, $x_{ij} \in R$, де приклад t_i характеризується m ознаками. Задано кількість кластерів $2 \leq g \leq N$ та параметр $w > 1$ - показник нечіткості, який показує розмитість кластерів. Розрахувати центри кластерів за формулою (нечіткий метод к-середніх):

$$c_k = \frac{\sum_{i=1}^{N} (u_{ki})^w \cdot t_i}{\sum_{i=1}^{N} (u_{ki})^w}, k = 1, ..., g,$$

де $U = \{(u_{ki})|k = 1, ..., g, i = 1, ..., N\}$ - випадковим чином задана матриця початкового розбиття, $u_{kj} \in [0, 1]$ - це ступінь належності j-ї точки k-му кластеру, де $u_{kj} = 1$ означає повну приналежність, $u_{kj} = 0.7$ означає приналежність до k-го кластеру зі ступенем 0.7, $\sum_{k=1}^{g} u_{ki} = 1$, $\sum_{i=1}^{N} u_{ki} < N$.

Перерахувати матрицю розбиття:

$$u_{ki} = \frac{1}{\sum_{v=1}^{g} \left(\frac{d^2(t_i, c_k)}{d^2(t_i, c_v)} \right)^{\frac{1}{w-1}}},$$

використати $d^2(t_i, c_k)$ - квадрат евклідової відстані між векторами.

Виконати декілька ітерацій з уточнення центрів кластерів.

20. Задано неорієнтовний граф J з V вершинами, де ваги дуг d_{ij} відомі для $\forall i, j = 1, ..., V$. Побудувати підграф G графу J за наступними етапами (алгоритм Борувки):

    1) Ініціалізувати граф $G := T$ з множиною ребер $E := \varnothing$.

    2) Поки G не зв'язний:

    - Ініціалізувати допоміжну множину ребер $U := \varnothing$.
    - Для кожної компоненти зв'язності графу G:
        - Ініціалізувати допоміжну множину ребер $S := \varnothing$.
        - Для кожної вершини вибраної компоненти зв'язності додати в S найкоротше ребро, яке поєднує цю вершину з якою-небудь вершиною другої компоненти.
        - Додати в U найкоротше ребро з S.
    - $E := E \cup U$.

    Надрукувати граф G.

21. Нехай n - кількість альтернатив моделей, m - кількість показників якості, за якими ці моделі оцінюються. Задано матрицю V значень пріоритетів або величин виконання альтернатив моделей за показниками, де $v_{i,j} \in R^+$ - пріоритет (величина виконання) i-ої моделі за j-м показником. Задано вектор нормованих ваг показників якості $w_j^q \in [0, 1]$, $\sum_{j=1}^{m} w_j^q = 1$.

    Розрахувати w_i^{aggr} агрегований пріоритет i-ої моделі, $i = 1, ..., n$ за множиною показників, використовуючи наступні формули.

    Дистрибутивний метод:

$$w_i^{aggr} = \sum_{j=1}^{m} r_{i,j} w_j^q,$$

де $r_{i,j} = \frac{v_{i,j}}{\sum_{k=1}^{n} v_{k,j}}$ для $\forall j = 1, .., m$.

Модифікований дистрибутивний метод:

$$v_i^{aggr} = \sum_{j=1}^{m} r_{i,j} w_j^q,$$

де $r_{i,j} = \frac{v_{i,j}}{\max_{k=1,...,n} v_{k,j}}$ для $\forall j = 1, .., m$.

Мультиплікативний метод:

$$v_i^{aggr} = \prod_{j=1}^{m} (v_{i,j})^{w_j^q}.$$

Метод на основі функції мінімуму:

$$v_i^{aggr} = \min_{j=1,...,m} v_{i,j} w_j^q.$$

Нормувати отримані пріоритети:

$$w_i^{aggr} = \frac{v_i^{aggr}}{\sum_{k=1}^{n} v_k^{aggr}}.$$

22. Нехай n - кількість альтернативних моделей, m - кількість показників якості, за якими ці моделі оцінюються. Задано матрицю W значень пріоритетів або величин виконання альтернатив моделей за показниками, де $w_{i,j} \in [0, 1]$ - нормований пріоритет (величина виконання) i-ої моделі за j-м показником так що $\sum_{i=1}^{n} w_{i,j} = 1$ для кожного $j = 1, ..., m$. Задано також вектор нормованих ваг показників якості $w_j^q \in [0, 1]$, $\sum_{j=1}^{m} w_j^q = 1$.

Знайти величини стійкості SI_j множини альтернатив моделей по кожному j-му показнику якості, $j = 1, ..., m$:

$$SI_j = \min_{i,k=1,...,n, i<k} (|\delta_{i,k,j}|),$$

$$\delta_{i,k,j} = \frac{w_k^{aggr} - w_i^{aggr}}{(w_{k,j} - w_{i,j}) w_j^q}$$

$$w_i^{aggr} = \sum_{j=1}^{m} w_{i,j} w_j^q.$$

Знайти величини стійкості $SI_{i,k}^{model}$ для кожної пари моделей (i, k) за множиною показників якості:

$$SI_{i,k}^{model} = \min_{j=1,...,m} (|\delta_{i,k,j}|), i, k = 1, ..., n,$$

де $\delta_{i,k,j}$ розраховуються за наведеною вище формулою.

23. Нехай n - кількість альтернативних моделей, m - кількість показників якості, за якими ці моделі оцінюються. Задано матрицю W значень пріоритетів або величин виконання альтернатив моделей за показниками, де $w_{i,j} \in [0,1]$ - нормований пріоритет (величина виконання) i-ої моделі за j-м показником так що $\sum_{i=1}^{n} w_{i,j} = 1$ для кожного $j = 1, ..., m$. Задано також вектор нормованих ваг показників якості $w_j^q \in [0,1]$, $\sum_{j=1}^{m} w_j^q = 1$.

    Зобразити графічно на одній координатній площині, див. приклад нижче:

    - значення пріоритетів $w_{i,j}$ моделей за показниками якості (критеріями рішень) - лініями різних кольорів, кожній альтернативі відповідає свій колір,
    - значення ваг показників якості (критеріїв) w_j^q стовпчиковою діаграмою на тій самій координатній площині,
    - агреговані пріоритети (глобальні ваги) кожної моделі $w_i^{aggr} = \sum_{j=1}^{m} w_{i,j} w_j^q$, $i = 1, ..., n$ точками відповідного кольору.

    Таким чином, умову і результати агрегування представляємо графічно для подальшого візуального аналізу чутливості.

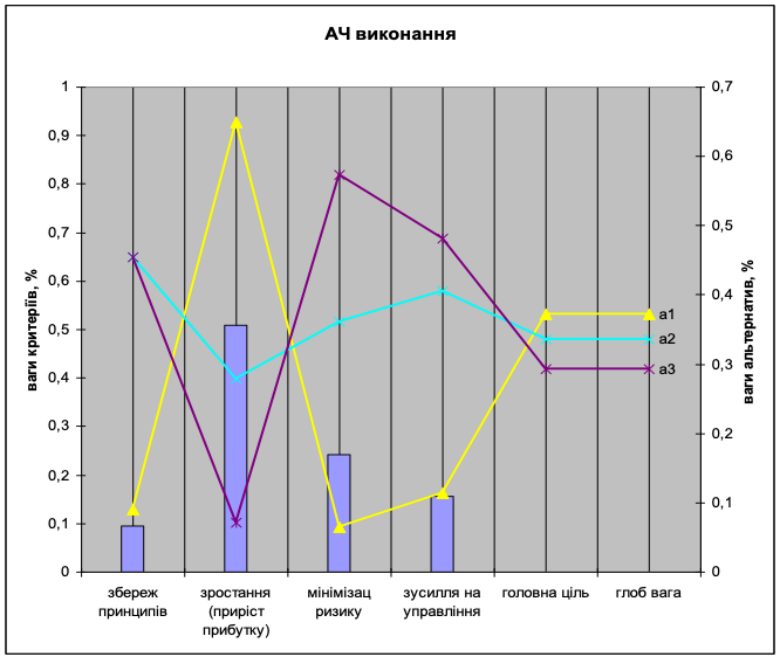

Рис. 1: Задача підтримки прийняття рішень: три альтернативи a1, a2, a3, чотири критерії рішень

24. Нехай n - кількість альтернативних моделей, m - кількість показників якості, за якими ці моделі оцінюються. Задано матрицю W значень пріоритетів або величин виконання альтернатив моделей за показниками,

де $w_{i,j} \in [0,1]$ - нормований пріоритет (величина виконання) i-ої моделі за j-м показником так що $\sum_{i=1}^{n} w_{i,j} = 1$ для кожного $j = 1, ..., m$. Задано також вектор нормованих ваг показників якості $w_j^q \in [0,1]$, $\sum_{j=1}^{m} w_j^q = 1$.

Зобразити графічно **для кожної пари** (i,k) альтернативних моделей у вигляді діаграми, див. приклад нижче:

- значення різниць пріоритетів $w_{i,j} - w_{k,j}$ обраних моделей за кожним j-м показником якості,
- значення різниці агрегованих пріоритетів (глобальних ваг) $w_i^{aggr} - w_k^{aggr}$ обраних моделей, $w_i^{aggr} = \sum_{j=1}^{m} w_{i,j} w_j^q$.

Таким чином, можемо візуально оцінити відмінності між виконанням двох обраних моделей за кожним показником якості, а також якими є відмінності у результатах - агрегованих пріоритетах цих двох моделей. Кількість рисунків з діаграмами дорівнює кількості пар альтернативних моделей.

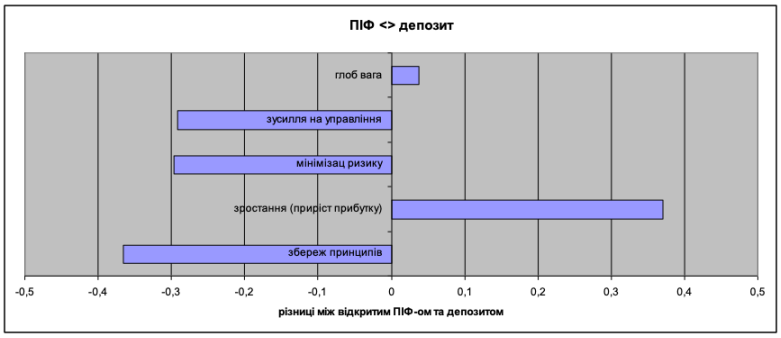

Рис. 2: Приклад різницевого аналізу чутливості для задачі підтримки прийняття рішень з чотирма критеріями рішень та двома альтернативами: ПІФ та депозит

25. Нехай n - кількість альтернативних моделей, m - кількість показників якості, за якими ці моделі оцінюються. Задано матрицю W значень пріоритетів або величин виконання альтернатив моделей за показниками, де $w_{i,j} \in [0,1]$ - нормований пріоритет (величина виконання) i-ої моделі за j-м показником так що $\sum_{i=1}^{n} w_{i,j} = 1$ для кожного $j = 1, ..., m$. Задано також вектор нормованих ваг показників якості $w_j^q \in [0,1]$, $\sum_{j=1}^{m} w_j^q = 1$.

    Розглянути по черзі кожний показник якості (критерій рішень). Нехай зафіксовано j-й критерій. Зобразити графічно залежності агрегованих пріоритетів (глобальних ваг) моделей від ваги цього j-го критерію, див. приклад нижче.

    (a) по осі абсцис вказується значення ваги w_j^{q*} j-го критерію з діапазону $[0,1]$, крок дискретизації 0.1 або 0.01,

(б) по осі ординат - значення агрегованих пріоритетів (глобальних ваг) кожної моделі: $w_i^{aggr} = \sum_{j=1}^{m} w_{i,j} w_j^{q*}$, $i = 1, ..., n$, що розраховані для встановленого на кроці (а) значення ваги w_j^{q*},

(в) червоною вертикальною лінією позначити реальне значення w_j^q ваги j-го показника якості.

Таким чином, ми відображаємо яким буде агрегований пріоритет кожної моделі, якщо б вага обраного показника якості (критерію) дорівнювала б нулю, 0.1, 0.2, ..., 0.9, 1.0. В подальшому ці графіки використовуються для градієнтного аналізу чутливості моделей. Зверніть увагу, що при зміні ваги критерію і встановлені її рівною w_j^{q*} на кроці (а), на наступному кроці (б) потрібно пропорційно перенормувати ваги всіх інших критеріїв для забезпечення умови $\sum_{j=1}^{m} w_j^q = 1$ перед розрахунком агрегованих значень.

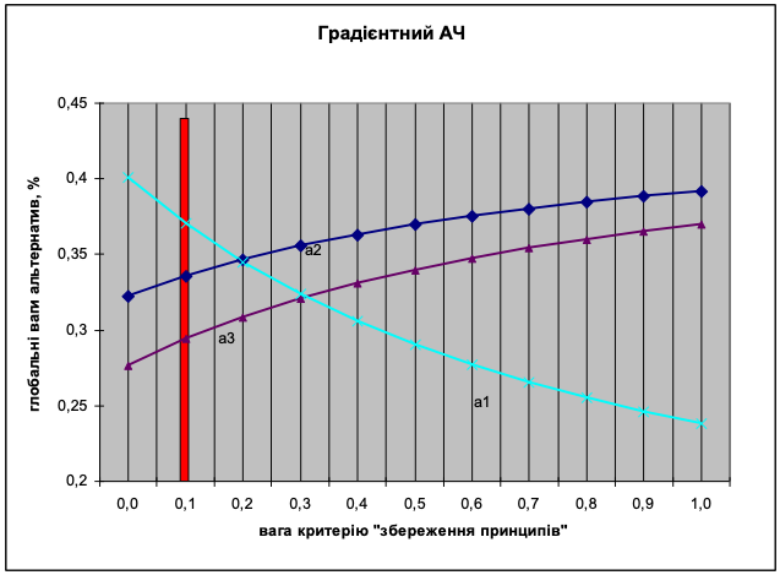

Рис. 3: Приклад градієнтного аналізу чутливості для задачі підтримки прийняття рішень з альтернативами a1, a2, a3

26. Задача аналізу ринкових кошиків. Задано множину товарів $I = \{i_1, i_2, ..., i_n\}$ та множину транзакцій $D = \{T_1, T_2, ..., T_m\}$, де $T = \{i_k | i_k \in I\} \subseteq I$ - транзакція - це множина товарів, які були куплені разом в одному чеку.

Підтримкою довільного набору $F \subseteq I$ називається число

$$Supp(F) = \frac{|D_F|}{|D|},$$

де D_F - множина транзакцій, які містять набір F:

$$D_F = \{T_j | F \subseteq T_j\},$$

$|D|$ - кількість елементів у множині D.

Знайти множину частих наборів товарів, використовуючи наступний алгоритм:

(а) Побудувати множину одноелементних частих наборів:

$$L_1 = \{i | i \in I, Supp(i) \geq Supp_{min}\},$$

де $Supp_{min}$ - заданий параметр - поріг мінімальної підтримки.

(б) Для всіх $k = 2, ..., n$:
Побудувати множини k-елементних частих наборів

$$L_k = \{F \cup \{i\} | F \in L_{k-1}, i \in L_1 \setminus F, Supp(F \cup \{i\}) \geq Supp_{min}\}.$$

(в) Якщо $L_k = \emptyset$, то вихід із циклу по k.

(г) $\{L_1 \cup L_2 \cup ... \cup L_k\}$ - результуюча множина частих наборів.

27. Задача аналізу ринкових кошиків. Задано множину товарів $I = \{i_1, i_2, ..., i_n\}$ та множину транзакцій $D = \{T_1, T_2, ..., T_m\}$, де $T = \{i_k | i_k \in I\} \subseteq I$ - транзакція - це множина товарів, які були куплені разом в одному чеку.

Перевести множину D до вертикального формату (TID-множини) S:

$$S = \{\{i, D_i\} | i \in I, D_i = \{T_j | i \in T_j\}\}.$$

Наприклад, $I = \{\text{хліб, масло, сік, вода, ковбаса}\}$.

Горизонтальний формат представлення множини D:

Номер транзакції	Товари
1	хліб, масло, сік
2	сік, вода
3	хліб, масло, ковбаса
4	хліб, масло, вода

Вертикальний формат представлення:

Номер транзакції	Товари
хліб	1, 3, 4
масло	1, 3, 4
сік	1, 2
вода	2, 4
ковбаса	3

28. Задача аналізу ринкових кошиків. Задано множину товарів $I = \{i_1, i_2, ..., i_n\}$ та S_1 - множину транзакцій у *вертикальному* форматі представлення (див. попередній варіант).

    Підтримкою довільного набору $F \subseteq I$ називається число

    $$Supp(F) = \frac{|D_F|}{|D|},$$

    де D_F - множина транзакцій, які містять набір F:

    $$D_F = \{T_j | F \subseteq T_j\},$$

    $|D|$ - кількість елементів у множині D.

    Знайти множину частих наборів товарів, використовуючи наступний алгоритм:

    (а) Побудувати множину одноелементних частих наборів:

    $$L_1 = \{i | i \in I, Supp(i) \geq Supp_{min}\},$$

    де $Supp_{min}$ - заданий параметр - поріг мінімальної підтримки.

    (б) Для всіх $k = 2, ..., n$:
    Побудувати кандидатів у множину k-елементних частих наборів:
    Для всіх $F_a, F_b \in L_{k-1}$:

    $$S_k = \{\{F^*, M^*_{F^*}\} | F^* = F_a \cup F_b, M^*_{F^*} = \{M_{F_a} \cap M_{F_b}\}, a < b, M_{F_a}, M_{F_b} \in S_{k-1}\}.$$

    Відбір частих наборів:

    $$L_k = \{F^* | F^* \in S_k, Supp(F^*) \geq Supp_{min}\}.$$

    (в) Якщо $L_k = \emptyset$ то вихід із циклу по k.

    (г) $\{L_1 \cup L_2 \cup ... \cup L_k\}$ - результуюча множина частих наборів.

2 Контрольні питання для захисту роботи

1. **Типи даних в Python.**

2. **Основи роботи в бібліотеці NumPy.**

    - Масиви NumPy:
        - Індексація масива. Доступ до окремих елементів багатовимірних масивів.
        - numpy.reshape. Навести приклади.
        - numpy.newaxis. Навести приклади.

  - Зрізи масивів: доступ до підмасивів.
  - Маскування з використанням булевих масивів.
  - numpy.concatenate. Навести приклади для одновимірного та двовимірного масивів.
  - numpy.vstack і numpy.hstack. Навести приклади.
  - numpy.split, numpy.hsplit, numpy.vsplit. Навести приклади.
  - Операція reduce. Навести приклади.
  - numpy.sum. Навести приклади.
  - numpy.prod. Навести приклади.
  - numpy.mean. Навести приклади.
  - numpy.var. Навести приклади.
  - numpy.amin, numpy.amax. Навести приклади.
- Універсальні функції над масивами в NumPy:
  - Поняття універсальної функції. Навіщо вони потрібні.
  - Арифметичні універсальні функції для масивів.
  - Правила транслювання (broadcasting).
  - Сортування масивів з використанням np.sort.
- Створення структурованих масивів в NumPy.

3. **Оперування даними за допомогою Pandas**

- Створення об'єкту Series бібліотеки Pandas.
- Об'єкт Series як словник.
- Об'єкт Series як одновимірний масив.
- Створення об'єкту DataFrame бібліотеки Pandas.
- Об'єкт DataFrame як словник.
- Об'єкт DataFrame як двовимірний масив.
- Застосування універсальних функцій до об'єктів Series і DataFrame.
- Застосування функцій агрегування до об'єктів Series і DataFrame.

4. **Візуалізація за допомогою Matplotlib**

- Побудова графіків із сценарію. Функція matplotlib.pyplot.show()
- Побудова графіків із блокноту IPython. Функція matplotlib.pyplot.plot().
- Побудова графіку функції $y = f(x)$ за допомогою matplotlib.pyplot.
- Налаштування кольору, стилю ліній, міток на графіках, легенди засобами matplotlib.pyplot.