

# ТЕСТУВАННЯ, ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОСНОВНИХ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ НАБОРІВ ЧИСЛОВИХ ДАНИХ

Виконав: Коломієць Микола

Група: ММШІ-1

25 травня 2025 р.

## Зміст

<b>1 Вступ</b>	<b>3</b>
1.1 Мета роботи . . . . .	3
1.2 Задачі дослідження . . . . .	3
<b>2 Теоретичні основи</b>	<b>3</b>
2.1 Алгоритм K-means . . . . .	3
2.2 Алгоритм Mean Shift . . . . .	4
2.3 Алгоритм FOREL . . . . .	5
<b>3 Методологія дослідження</b>	<b>5</b>
3.1 Типи наборів даних . . . . .	5
3.2 Метрики оцінювання . . . . .	5
3.2.1 Продуктивність . . . . .	5
3.2.2 Якість кластеризації . . . . .	6
3.2.3 Стабільність . . . . .	6
<b>4 Експериментальні результати</b>	<b>6</b>
4.1 Результати продуктивності . . . . .	6
4.2 Результати аналізу стабільності . . . . .	7
<b>5 Аналіз результатів</b>	<b>7</b>
5.1 Продуктивність . . . . .	7
5.1.1 Час виконання . . . . .	7
5.1.2 Використання пам'яті . . . . .	7
5.2 Аналіз стабільності . . . . .	8
5.3 Якість кластеризації за типами даних . . . . .	8

<b>6 Висновки та рекомендації</b>	<b>8</b>
6.1 Основні висновки . . . . .	8
6.2 Практичні рекомендації . . . . .	9
6.2.1 Критичні попередження . . . . .	9
6.3 Загальний рейтинг алгоритмів . . . . .	9
<b>A Код реалізації алгоритмів</b>	<b>9</b>
A.1 K-means . . . . .	9
A.2 Mean Shift . . . . .	10
A.3 FOREL . . . . .	11
<b>Б Візуалізації</b>	<b>12</b>
B.1 Порівняння результатів кластеризації . . . . .	12
B.1.1 Кругові кластери . . . . .	12
B.1.2 Еліптичні кластери . . . . .	13
B.1.3 Літерні форми . . . . .	14
B.1.4 Змішані типи . . . . .	15
B.2 Аналіз стабільності . . . . .	16
B.2.1 Стабільність K-means . . . . .	16
B.2.2 Стабільність FOREL . . . . .	17
B.2.3 Стабільність Mean Shift . . . . .	18
B.3 Порівняльні діаграми . . . . .	19
B.3.1 Продуктивність . . . . .	19

# 1 Вступ

Кластеризація є одним з основних методів аналізу даних у машинному навчанні та статистиці. Це процес групування об'єктів таким чином, щоб об'єкти всередині однієї групи (кластера) були більш схожими між собою, ніж з об'єктами з інших груп.

## 1.1 Мета роботи

Метою даної лабораторної роботи є:

- Дослідження та порівняння ефективності трьох алгоритмів кластеризації: K-means, Mean Shift та FOREL
- Аналіз поведінки алгоритмів на різних типах наборів даних
- Дослідження стабільності (неперервності) алгоритмів
- Порівняння продуктивності та використання ресурсів

## 1.2 Задачі дослідження

1. Реалізація трьох алгоритмів кластеризації
2. Генерація різних типів наборів даних для тестування
3. Проведення порівняльного аналізу якості кластеризації
4. Дослідження стабільності алгоритмів при малих змінах у даних
5. Вимірювання та порівняння продуктивності
6. Візуалізація результатів

# 2 Теоретичні основи

## 2.1 Алгоритм K-means

K-means є одним з найпопулярніших алгоритмів кластеризації. Він розділяє дані на  $k$  кластерів, мінімізуючи суму квадратів відстаней від точок до центрів кластерів.

---

### Algorithm 1 Алгоритм K-means

---

**Require:** Набір точок  $X = \{x_1, x_2, \dots, x_n\}$ , кількість кластерів  $k$

**Ensure:** Центри кластерів  $C = \{c_1, c_2, \dots, c_k\}$ , мітки точок  $L = \{l_1, l_2, \dots, l_n\}$

- 1: Ініціалізувати  $k$  центрів випадковим чином
  - 2: **repeat**
  - 3:   **for** кожної точки  $x_i$  **do**
  - 4:     Призначити  $x_i$  до найближчого центру
  - 5:   **end for**
  - 6:   **for** кожного кластера  $j$  **do**
  - 7:     Оновити центр  $c_j$  як середнє арифметичне точок кластера
  - 8:   **end for**
  - 9: **until** центри не змінюються
-

### **Переваги:**

- Простота реалізації
- Швидкість роботи
- Добре працює з круговими кластерами

### **Недоліки:**

- Необхідність заздалегідь знати кількість кластерів
- Чутливість до початкової ініціалізації
- Погано працює з кластерами неправильної форми

## **2.2 Алгоритм Mean Shift**

Mean Shift — це неієрархічний алгоритм кластеризації, який знаходить моди (локальні максимуми) в розподілі щільності точок.

---

### **Algorithm 2** Алгоритм Mean Shift

---

**Require:** Набір точок  $X = \{x_1, x_2, \dots, x_n\}$ , bandwidth  $h$

**Ensure:** Мітки точок  $L = \{l_1, l_2, \dots, l_n\}$

- 1: **for** кожної точки  $x_i$  **do**
  - 2:    $current\_point \leftarrow x_i$
  - 3:   **repeat**
  - 4:     Знайти точки в межах bandwidth від  $current\_point$
  - 5:      $new\_point \leftarrow$  середнє арифметичне точок в межах bandwidth
  - 6:      $current\_point \leftarrow new\_point$
  - 7:   **until** збіжність
  - 8:   Зберегти фінальну позицію для  $x_i$
  - 9: **end for**
  - 10: Групувати точки за їх фінальними позиціями
- 

### **Переваги:**

- Автоматичне визначення кількості кластерів
- Працює з кластерами довільної форми
- Стійкість до викидів

### **Недоліки:**

- Необхідність налаштування bandwidth
- Більш повільний за K-means
- Чутливість до вибору bandwidth

## 2.3 Алгоритм FOREL

FOREL (FORmal EElement) — алгоритм кластеризації, що використовує концепцію формальних елементів для знаходження кластерів.

---

### Algorithm 3 Алгоритм FOREL

---

**Require:** Набір точок  $X = \{x_1, x_2, \dots, x_n\}$ , bandwidth  $r$

**Ensure:** Мітки точок  $L = \{l_1, l_2, \dots, l_n\}$

```
1: for кожної непризначеної точки  $x_i$  do
2:    $center \leftarrow x_i$ 
3:   repeat
4:     Знайти точки в межах  $r$  від  $center$ 
5:      $new\_center \leftarrow$  середнє арифметичне цих точок
6:      $center \leftarrow new\_center$ 
7:   until збіжність
8:   Призначити всі точки в межах  $r$  від фінального центру до нового кластера
9: end for
```

---

#### Переваги:

- Автоматичне визначення кількості кластерів
- Стійкість до форми кластерів
- Добре обробляє кластери різної щільності

#### Недоліки:

- Залежність від параметра bandwidth
- Можливе утворення кластерів, що перекриваються

## 3 Методологія дослідження

### 3.1 Типи наборів даних

Для тестування алгоритмів було створено п'ять типів наборів даних:

1. **Кругові кластери** — класичні кластери у формі кіл з гауссівським розподілом
2. **Еліптичні кластери** — витягнуті кластери з різними орієнтаціями
3. **Прямокутні кластери** — кластери у формі прямокутників
4. **Літерні форми** — кластери у формі літер (C, L, T)
5. **Змішані типи** — комбінація різних геометричних форм

### 3.2 Метрики оцінювання

#### 3.2.1 Продуктивність

- **Час виконання** — час, необхідний для виконання алгоритму
- **Використання пам'яті** — об'єм оперативної пам'яті під час виконання

### 3.2.2 Якість кластеризації

- Кількість знайдених кластерів — порівняння з очікуваною кількістю
- Повнота призначення — частка точок, що були призначені до кластерів
- Візуальна оцінка — якість розділення кластерів

### 3.2.3 Стабільність

Для оцінки стабільності проводилися  $\delta$ -збурення:

- Випадковий зсув однієї точки на відстань  $\delta$
- Порівняння результатів до і після збурення
- Обчислення коефіцієнта схожості результатів

Коефіцієнт схожості обчислювався як:

$$S = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(l_i^{orig} = l_i^{pert})$$

де  $l_i^{orig}$  та  $l_i^{pert}$  — мітки точки  $i$  до і після збурення відповідно (у k-means номер кластеру може змінюватись, на це була зроблена поправка).

## 4 Експериментальні результати

### 4.1 Результати продуктивності

Табл. 1: Порівняння продуктивності алгоритмів

Алгоритм	Набір даних	Час (сек)	Пам'ять (МБ)	Параметр
K-means	Кругові кластери	0.0024	0.07	k=2
	Еліптичні кластери	0.0029	0.14	k=4
	Прямокутні кластери	0.0043	0.07	k=3
	Літерні форми	0.0017	0.17	k=3
	Змішані типи	0.0023	0.14	k=4
FOREL	Кругові кластери	0.0301	0.39	bw=2.0
	Еліптичні кластери	0.0188	0.30	bw=3.0
	Прямокутні кластери	0.0044	0.15	bw=2.5
	Літерні форми	0.0235	0.37	bw=4.0
	Змішані типи	0.0150	0.23	bw=3.0
Mean Shift	Кругові кластери	0.1029	0.45	bw=2.0
	Еліптичні кластери	0.0864	0.40	bw=3.0
	Прямокутні кластери	0.0708	0.37	bw=2.5
	Літерні форми	0.1006	0.49	bw=4.0
	Змішані типи	0.0962	0.38	bw=3.0

**Примітка:** k — кількість кластерів для K-means, bw — bandwidth для FOREL та Mean Shift.

## 4.2 Результати аналізу стабільності

Табл. 2: Аналіз стабільності алгоритмів

<b>Алгоритм</b>	<b>Набір даних</b>	$\delta = 0.2$	$\delta = 2.0$	$\delta = 4.0$
K-means	Кругові кластери	1.000	1.000	1.000
	Еліптичні кластери	0.903	0.903	0.410
	Прямокутні кластери	1.000	1.000	1.000
	Літерні форми	0.877	0.187	0.477
	Змішані типи	0.236	0.340	0.278
FOREL	Кругові кластери	1.000	0.993	0.997
	Еліптичні кластери	1.000	1.000	0.580
	Прямокутні кластери	1.000	1.000	0.997
	Літерні форми	0.997	0.997	1.000
	Змішані типи	1.000	1.000	0.998
Mean Shift	Кругові кластери	1.000	0.997	0.997
	Еліптичні кластери	1.000	1.000	0.747
	Прямокутні кластери	1.000	1.000	1.000
	Літерні форми	1.000	1.000	1.000
	Змішані типи	0.835	0.795	0.722

**Примітка:** Значення 1.000 означає повну стабільність (100% схожості результатів), 0.000 — повну нестабільність.

## 5 Аналіз результатів

### 5.1 Продуктивність

#### 5.1.1 Час виконання

Аналіз показав наступну ієрархію за швидкістю виконання:

1. **K-means** — найшвидший (0.0017-0.0043 сек)
2. **FOREL** — середній (0.0044-0.0301 сек)
3. **Mean Shift** — найповільніший (0.0708-0.1029 сек)

#### 5.1.2 Використання пам'яті

Ефективність використання пам'яті також корелює зі складністю алгоритмів:

- **K-means:** 0.07-0.17 МБ — найефективніший
- **FOREL:** 0.15-0.39 МБ — помірне споживання
- **Mean Shift:** 0.37-0.49 МБ — найбільше споживання

K-means використовує в 2-7 разів менше пам'яті порівняно з іншими алгоритмами, що робить його ідеальним для обробки великих наборів даних.

## 5.2 Аналіз стабільності

Результати тестування стабільності виявили кардинальні відмінності між алгоритмами:

- K-means показав катастрофічно низьку стабільність відносно ініціалізації, проте при правильно підібраному методу ініціалізації він дуже стабільний відносно  $\delta$ -збурень.
- FOREL демонструє найвищу стабільність.
- Mean Shift показує дуже хорошу стабільність.

## 5.3 Якість кластеризації за типами даних

- **Кругові кластери:** Рейтинг для кругових кластерів: K-means > FOREL = Mean Shift
- **Еліптичні кластери:** Рейтинг для еліптичних кластерів: Mean Shift > FOREL > K-means - під час тестування K-means неправильно ідентифікував кластери і два кластери неправильно розділились.
- **Складні форми (літерні):** Рейтинг для складних форм: Mean Shift = FOREL = K-means - всі методи показали себе не ефективними.
- **Змішані типи:** Рейтинг для змішаних типів: FOREL > Mean Shift > K-means (знову K-means неправильно ініціалізувався, при правильній ініціалізації результати найкращі).
- **Прямоугутні кластери:** Всі впоралися ідеально.

# 6 Висновки та рекомендації

## 6.1 Основні висновки

Експериментальне дослідження виявило кардинальні відмінності між алгоритмами:

1. **Продуктивність vs Стабільність:** Виявлено обернену залежність між швидкістю та стабільністю. K-means, будучи найшвидшим, демонструє критично низьку стабільність.
2. **FOREL — оптимальний баланс:** FOREL показує найкращий компроміс між продуктивністю та стабільністю, забезпечуючи виняткову стабільність (0.993-1.000) при помірній швидкості.
3. **Mean Shift — стабільність з ціною:** Mean Shift забезпечує високу стабільність, але за рахунок значного зниження продуктивності (найповільніший у 2-42 рази).
4. **K-means — проблематична нестабільність:** K-means показав низьку стабільність.

5. **Тип даних має значення:** Складність набору даних драматично впливає на поведінку алгоритмів. Змішані типи є особливо проблематичними для K-means та частково для Mean Shift.

## 6.2 Практичні рекомендації

Табл. 3: Рекомендації щодо вибору алгоритму за критеріями

Пріоритет/Сценарій	Рекомендований алгоритм
Максимальна швидкість	K-means (з обережністю)
Стабільність критична	FOREL або Mean Shift
Збалансованість	FOREL
Складні форми кластерів	Mean Shift
Великі обсяги даних	FOREL (компроміс)
Прототипування/експерименти	K-means
Продукційні системи	FOREL або Mean Shift
Кругові/еліптичні кластери	FOREL
Невідома кількість кластерів	FOREL або Mean Shift

### 6.2.1 Критичні попередження

- **K-means не рекомендується** для задач, де стабільність результатів є критичною
- **Складні кластери** вимагають особливої уваги при виборі алгоритму
- **Mean Shift** може бути неефективним для великих наборів даних через низьку швидкість
- Параметр **bandwidth** у FOREL та Mean Shift потребує ретельного налаштування

## 6.3 Загальний рейтинг алгоритмів

Враховуючи всі критерії (продуктивність, стабільність, універсальність):

1. **FOREL** — оптимальний вибір для більшості задач
2. **Mean Shift** — коли стабільність важливіша за швидкість
3. **K-means** — тільки для швидкого прототипування або коли стабільність не критична або, якщо є можливість декілька разів запускати алгоритм та підлаштовувати методі ініціалізації.

## A Код реалізації алгоритмів

### A.1 K-means

Лістинг 1: Реалізація K-means

```
def k_means(dots: list, k: int, max_iter: int = 100):
    dots = np.array(dots)
    centers = dots[np.random.choice(dots.shape[0], k, replace=False)]
    
    for i in range(max_iter):
        distances = np.linalg.norm(dots[:, np.newaxis] - centers,
                                     axis=2)
        labels = np.argmin(distances, axis=1)

        new_centers = np.array([dots[labels == j].mean(axis=0)
                               for j in range(k)])
        
        if np.all(centers == new_centers):
            break

    centers = new_centers

return centers, labels
```

## A.2 Mean Shift

Лістинг 2: Реалізація Mean Shift

```
def mean_shift(dots: list, bandwidth: float, max_iter: int = 300):
    dots = np.array(dots)
    n_samples, n_features = dots.shape
    final_positions = np.zeros_like(dots)

    for i in range(n_samples):
        current_point = dots[i].copy()

        for iteration in range(max_iter):
            distances = np.linalg.norm(dots - current_point, axis
                                         =1)
            in_bandwidth = dots[distances <= bandwidth]

            if len(in_bandwidth) <= 1:
                break

            new_point = np.mean(in_bandwidth, axis=0)

            if np.allclose(current_point, new_point, atol=1e-4):
                break

            current_point = new_point

        final_positions[i] = current_point
```

```

cluster_threshold = bandwidth * 0.3
labels = np.full(n_samples, -1)
cluster_id = 0

for i in range(n_samples):
    if labels[i] != -1:
        continue

    current_center = final_positions[i]
    distances_to_center = np.linalg.norm(
        final_positions - current_center, axis=1)

    similar_points = distances_to_center <= cluster_threshold
    labels[similar_points] = cluster_id
    cluster_id += 1

return labels

```

### A.3 FOREL

Лістинг 3: Реалізація FOREL

```

def forel(dots: list, bandwidth: float, max_iter: int = 1000):
    dots = np.array(dots)
    n_samples, n_features = dots.shape
    labels = np.full(n_samples, -1)
    cluster_centers = []
    cluster_id = 0

    for i in range(n_samples):
        if labels[i] != -1:
            continue

        current_center = dots[i].copy()
        prev_center = None

        for iteration in range(max_iter):
            distances = np.linalg.norm(dots - current_center, axis
                =1)
            points_in_sphere = distances <= bandwidth

            if not np.any(points_in_sphere):
                break

            points_in_bandwidth = dots[points_in_sphere]
            new_center = np.mean(points_in_bandwidth, axis=0)

            if (prev_center is not None and
                np.allclose(new_center, prev_center, atol=1e-6)):
                break

            cluster_centers.append(new_center)
            labels[points_in_sphere] = cluster_id
            cluster_id += 1

```

```

    prev_center = current_center.copy()
    current_center = new_center

    final_distances = np.linalg.norm(dots - current_center,
        axis=1)
    points_to_assign = final_distances <= bandwidth

    unassigned_mask = labels == -1
    assignment_mask = points_to_assign & unassigned_mask

    if np.any(assignment_mask):
        labels[assignment_mask] = cluster_id
        cluster_centers.append(current_center)
        cluster_id += 1

    return labels

```

## Б Візуалізації

### Б.1 Порівняння результатів кластеризації

Нижче представлені ключові візуалізації, що демонструють роботу алгоритмів на різних типах даних.

#### Б.1.1 Кругові кластери



Рис. 1: Результати K-means на кругових кластерах



Рис. 2: Результати FOREL на кругових кластерах

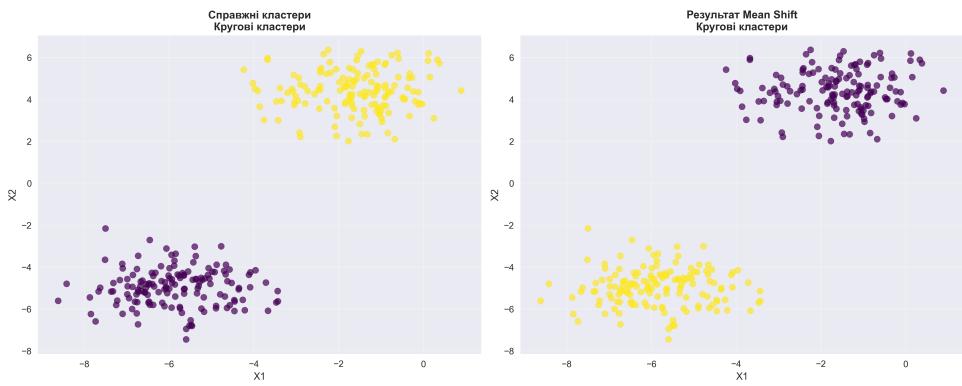


Рис. 3: Результати Mean Shift на кругових кластерах

### Б.1.2 Еліптичні кластери



Рис. 4: Результати K-means на еліптичних кластерах

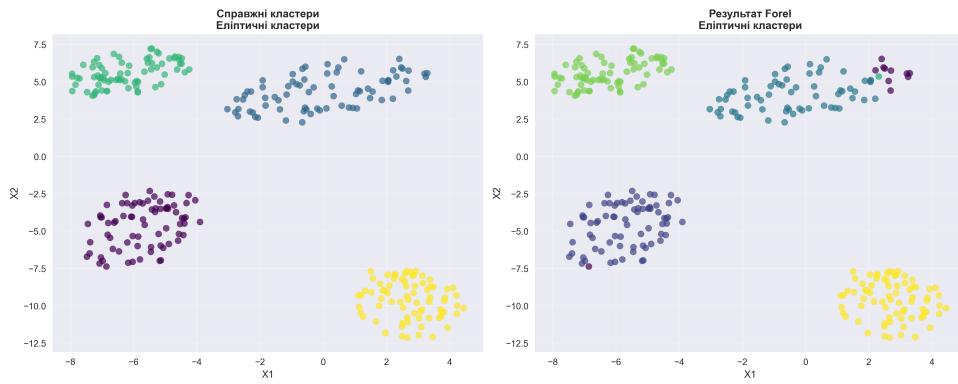


Рис. 5: Результати FOREL на еліптичних кластерах

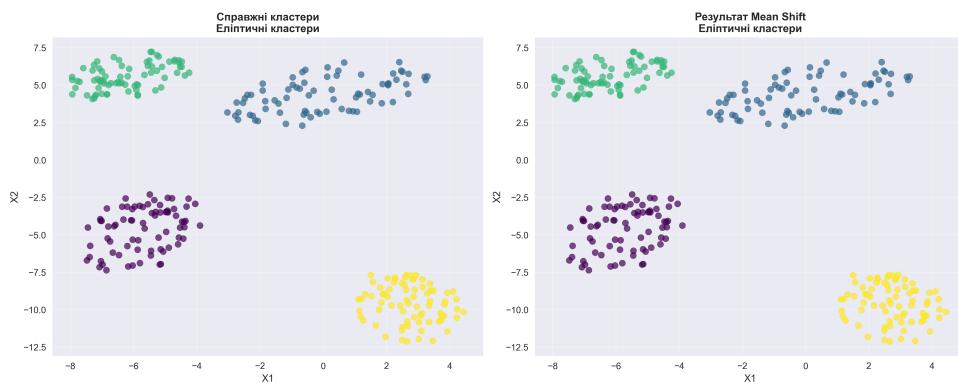


Рис. 6: Результати Mean Shift на еліптичних кластерах

### Б.1.3 Літерні форми

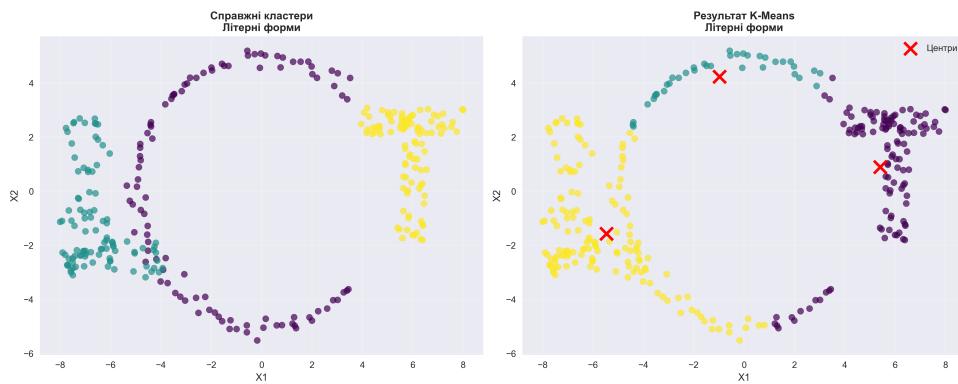


Рис. 7: Результати K-means на кластерах у формі літер

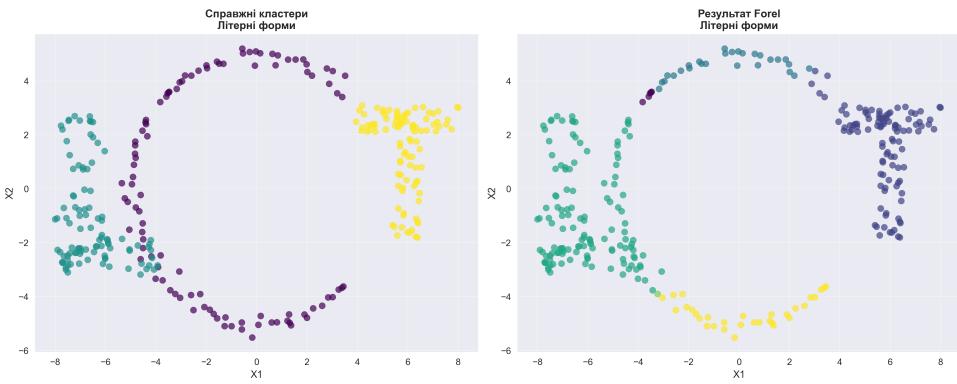


Рис. 8: Результати FOREL на кластерах у формі літер

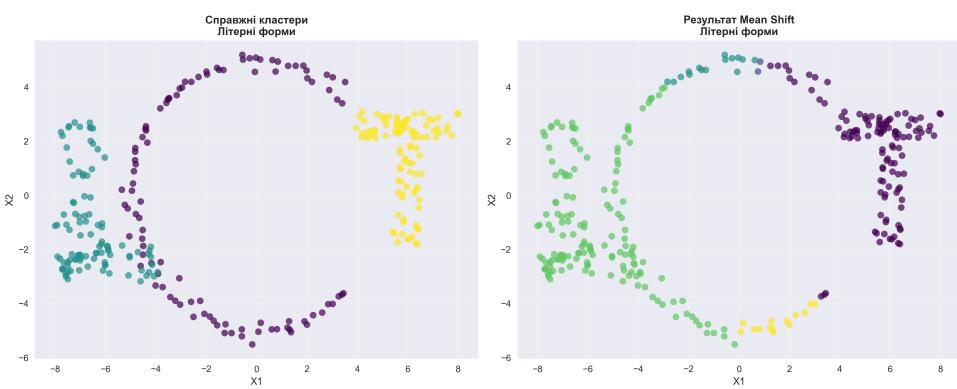


Рис. 9: Результати Mean Shift на кластерах у формі літер

#### Б.1.4 Змішані типи

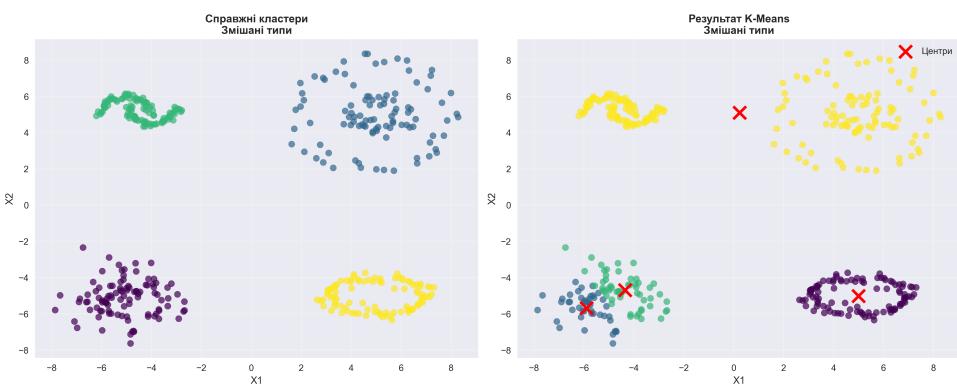


Рис. 10: Результати K-means на змішаних типах кластерів

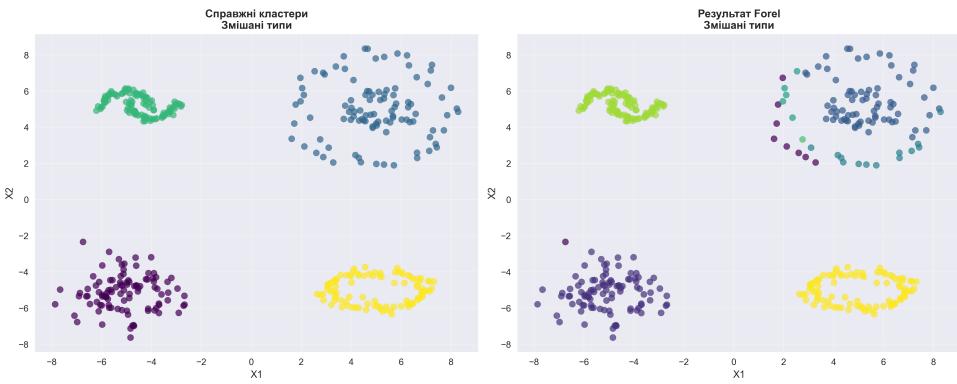


Рис. 11: Результати FOREL на змішаних типах кластерів

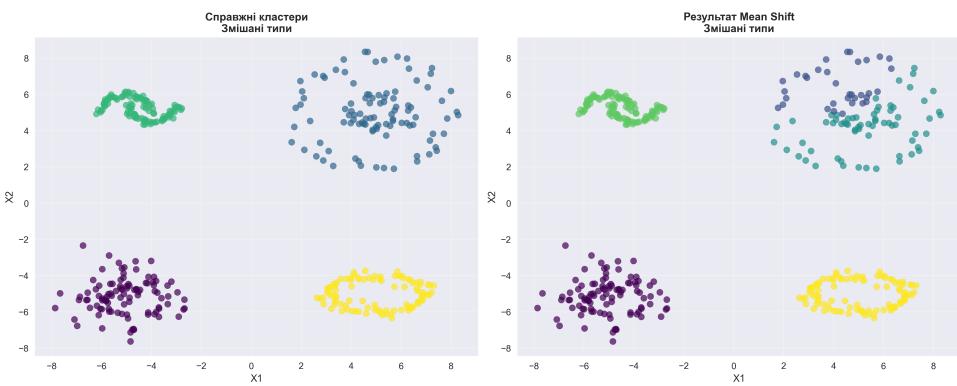


Рис. 12: Результати Mean Shift на змішаних типах кластерів

## Б.2 Аналіз стабільності

### Б.2.1 Стабільність K-means

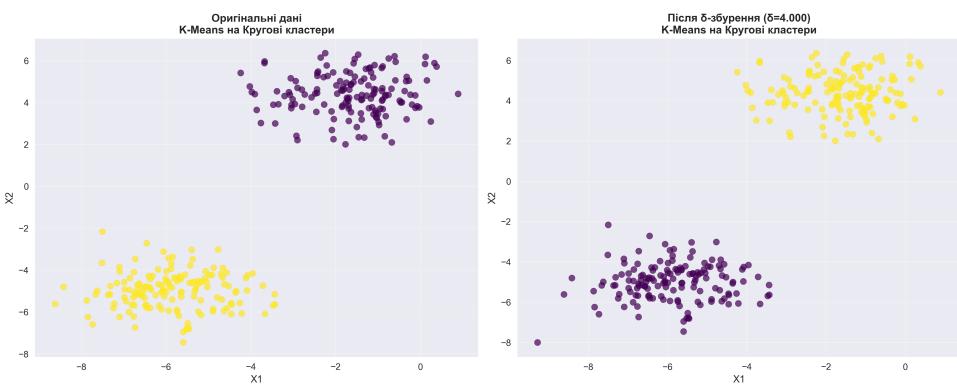


Рис. 13: Аналіз стабільності K-means на кругових кластерах

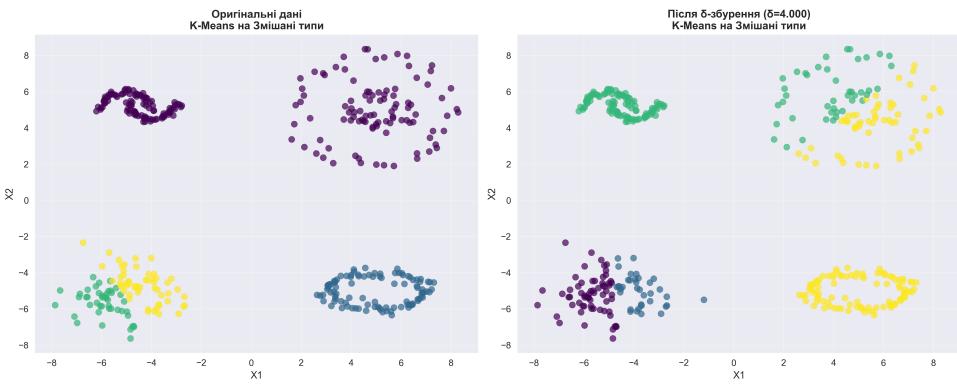


Рис. 14: Аналіз стабільності K-means на змішаних типах (критична нестабільність)

### Б.2.2 Стабільність FOREL

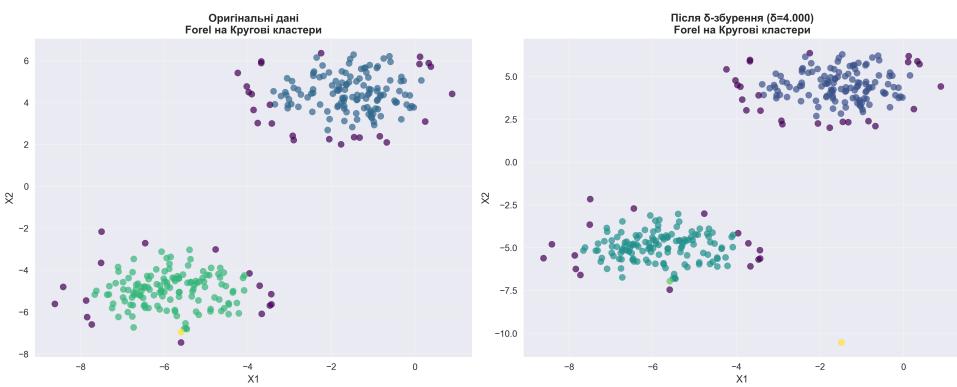


Рис. 15: Аналіз стабільності FOREL на кругових кластерах

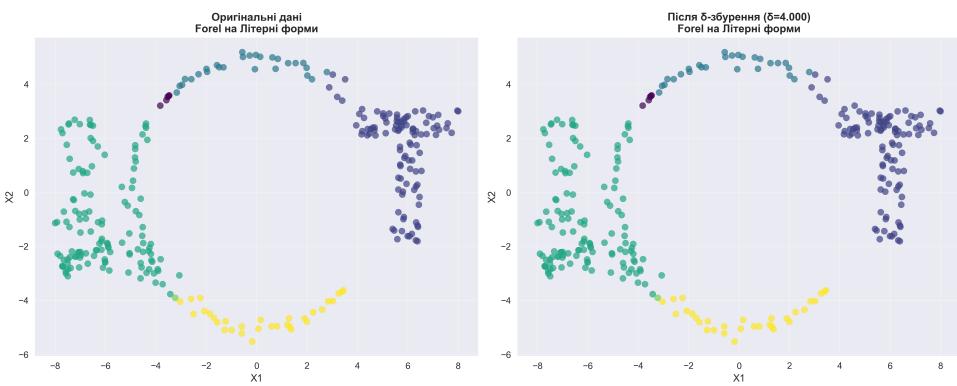


Рис. 16: Аналіз стабільності FOREL на літерних формах

### Б.2.3 Стабільність Mean Shift

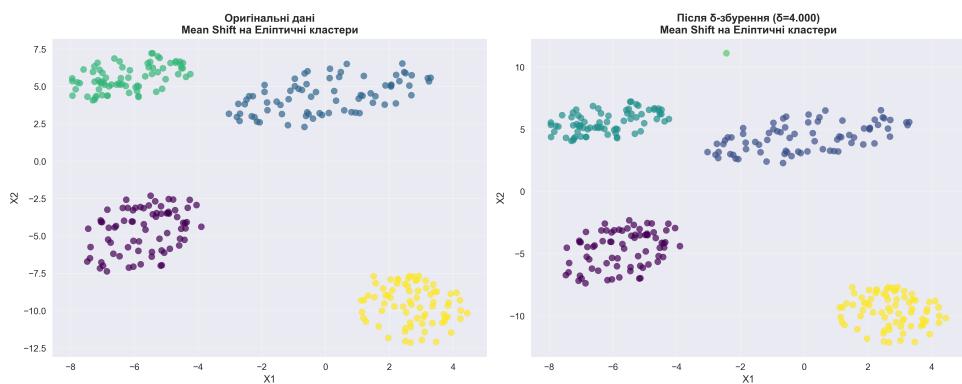


Рис. 17: Аналіз стабільності Mean Shift на еліптичних кластерах

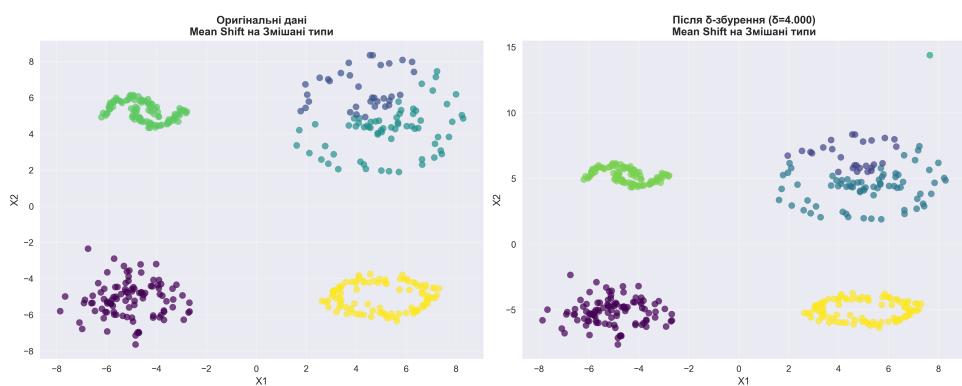


Рис. 18: Аналіз стабільності Mean Shift на змішаних типах

## Б.3 Порівняльні діаграми

### Б.3.1 Продуктивність

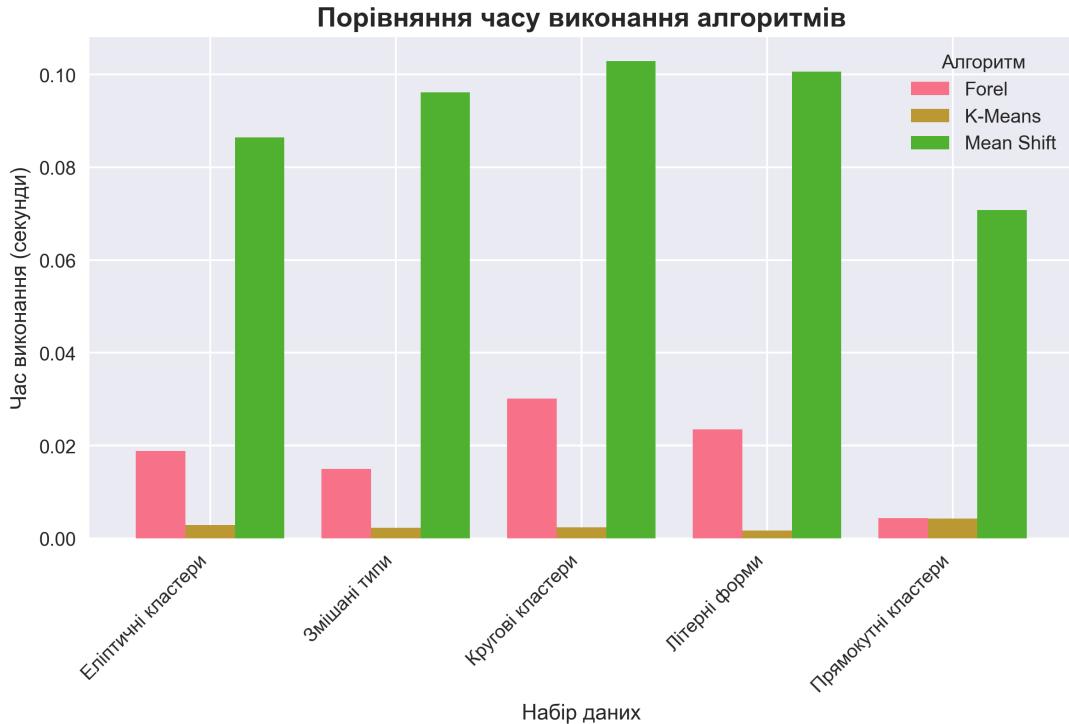


Рис. 19: Порівняння часу виконання алгоритмів на різних типах даних

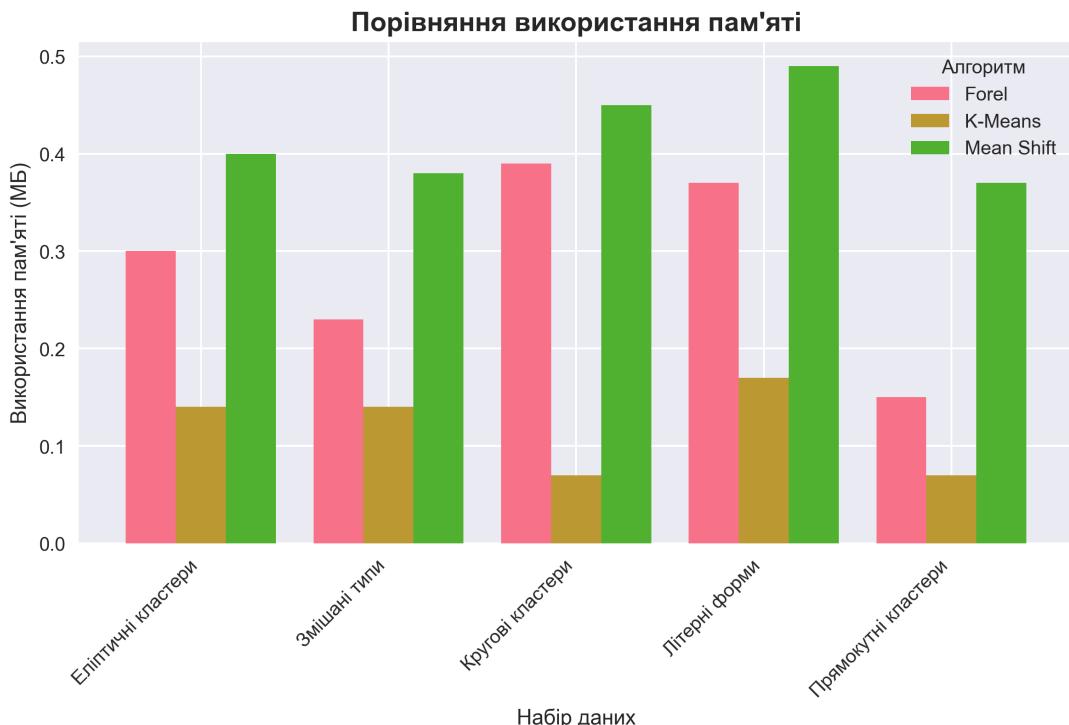


Рис. 20: Порівняння використання пам'яті алгоритмами