Виконав студент групи ПДМ-51
Державного університету
інформаційно-комунікаційних технологій
Скворцов Михайло Олександрович

Лабораторна №4

Методи неконтрольованого навчання Кластеризація

Код до всіх завдань можна знайти за посиланням:
https://github.com/Mike-Skvortsov/IAD

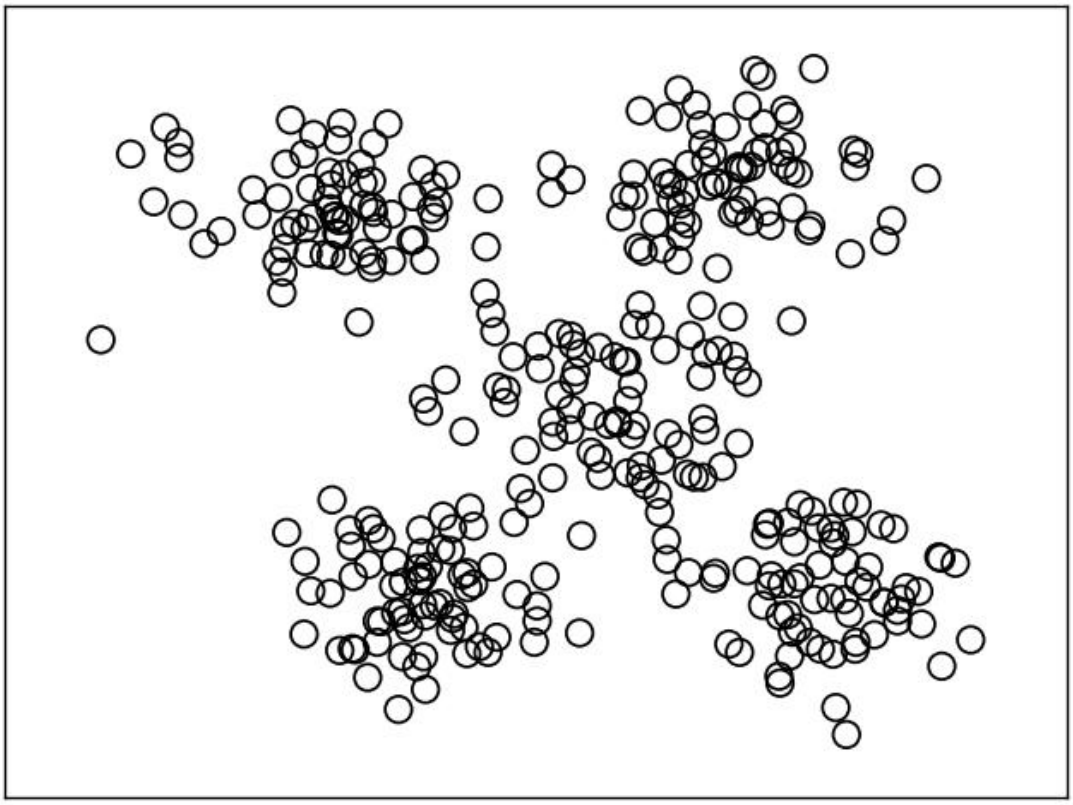
Завдання 2.1. Кластеризація даних за допомогою методу k-середніх

У ході виконання лабораторної роботи було проведено кластеризацію 2D-даних методом k-середніх. Алгоритм успішно розподілив дані на 5 підгруп відповідно до подібності. Центроїди кластерів були знайдені автоматично, що забезпечило оптимальне розбиття. Візуалізація показала чітку межу між групами, що підтверджує коректність роботи алгоритму.

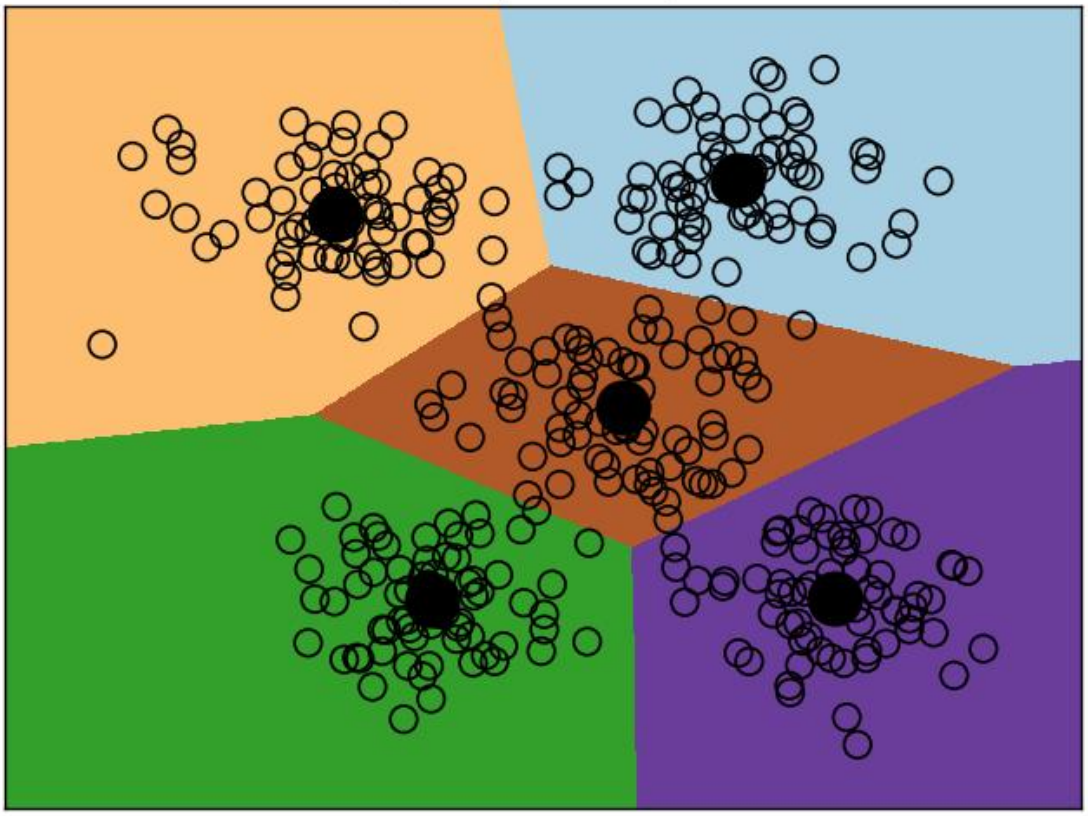
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

X = np.loadtxt('data_clustering.txt', delimiter=',')

num_clusters = 5

plt.figure()
plt.scatter(X[:,0], X[:,1], marker='o',
facecolors='none',
            edgecolors='black', s=80)
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
plt.title('Вхідні дані')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()

kmeans = KMeans(init='k-means++',
n_clusters=num_clusters, n_init=10)

kmeans.fit(X)
```

```
step_size = 0.01

x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
x_vals, y_vals = np.meshgrid(np.arange(x_min, x_max,
step_size),
                             np.arange(y_min, y_max,
step_size))

output = kmeans.predict(np.c_[x_vals.ravel(),
y_vals.ravel()])

output = output.reshape(x_vals.shape)
plt.figure()
plt.imshow(output, interpolation='nearest',
           extent=(x_vals.min(), x_vals.max(),
                   y_vals.min(), y_vals.max()),
           cmap=plt.cm.Paired,
           aspect='auto',
           origin='lower')

plt.scatter(X[:,0], X[:,1], marker='o',
facecolors='none',
            edgecolors='black', s=80)

cluster_centers = kmeans.cluster_centers_
plt.scatter(cluster_centers[:,0], cluster_centers[:,1],
            marker='o', s=210, linewidths=4,
color='black',
            zorder=12, facecolors='black')

x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
plt.title("Границя кластерів")
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```

Вхідні дані



Границя кластерів



У ході виконання лабораторної роботи було проведено кластеризацію двовимірного набору даних за допомогою методу k-середніх (k-means). Було задано кількість кластерів k=5, яка попередньо визначалась візуальною оцінкою розподілу даних.

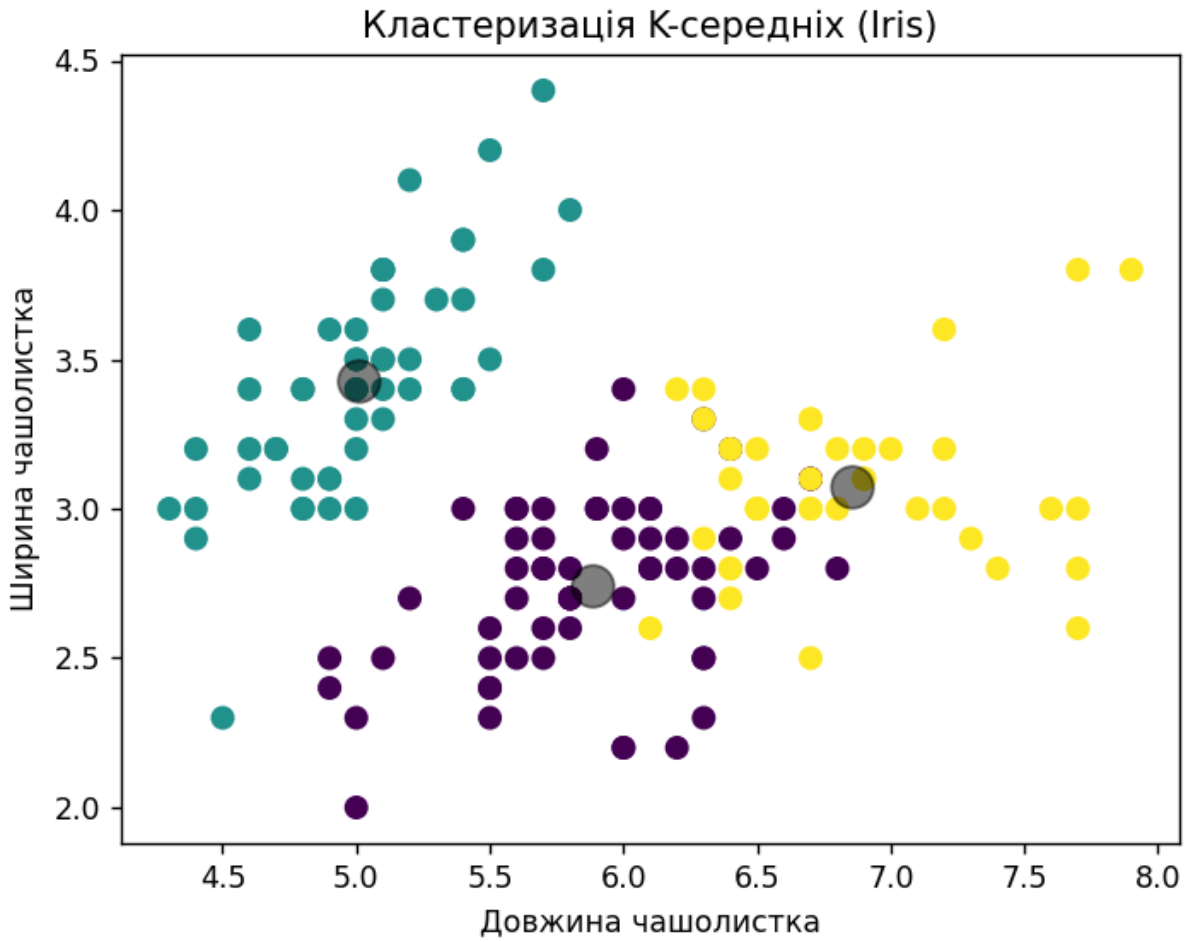
Для ініціалізації центроїдів використовувався вдосконалений метод k-means++, що забезпечив швидку збіжність алгоритму. В результаті навчання модель автоматично визначила положення центрів кластерів (центроїдів), які найкраще описують підгрупи точок у просторі.

На візуалізації чітко видно розподіл даних на окремі кластери з кольоровими областями та центрами кластерів. Це підтверджує ефективність обраного методу кластеризації та правильність реалізації алгоритму.

Отже, поставлена мета виконана — дані успішно кластеризовано, виявлено приховану структуру, і результати подано у наочній графічній формі.

Завдання 2.2. Кластеризація K-середніх для набору даних Iris

```
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans

iris = load_iris()
X = iris['data']
y = iris['target']

kmeans = KMeans(n_clusters=3, random_state=0)

kmeans.fit(X)

y_kmeans = kmeans.predict(X)

plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50,
cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black',
s=200, alpha=0.5)

plt.title("Кластеризація K-середніх (Iris)")
plt.xlabel("Довжина чашолистка")
plt.ylabel("Ширина чашолистка")
plt.show()
```



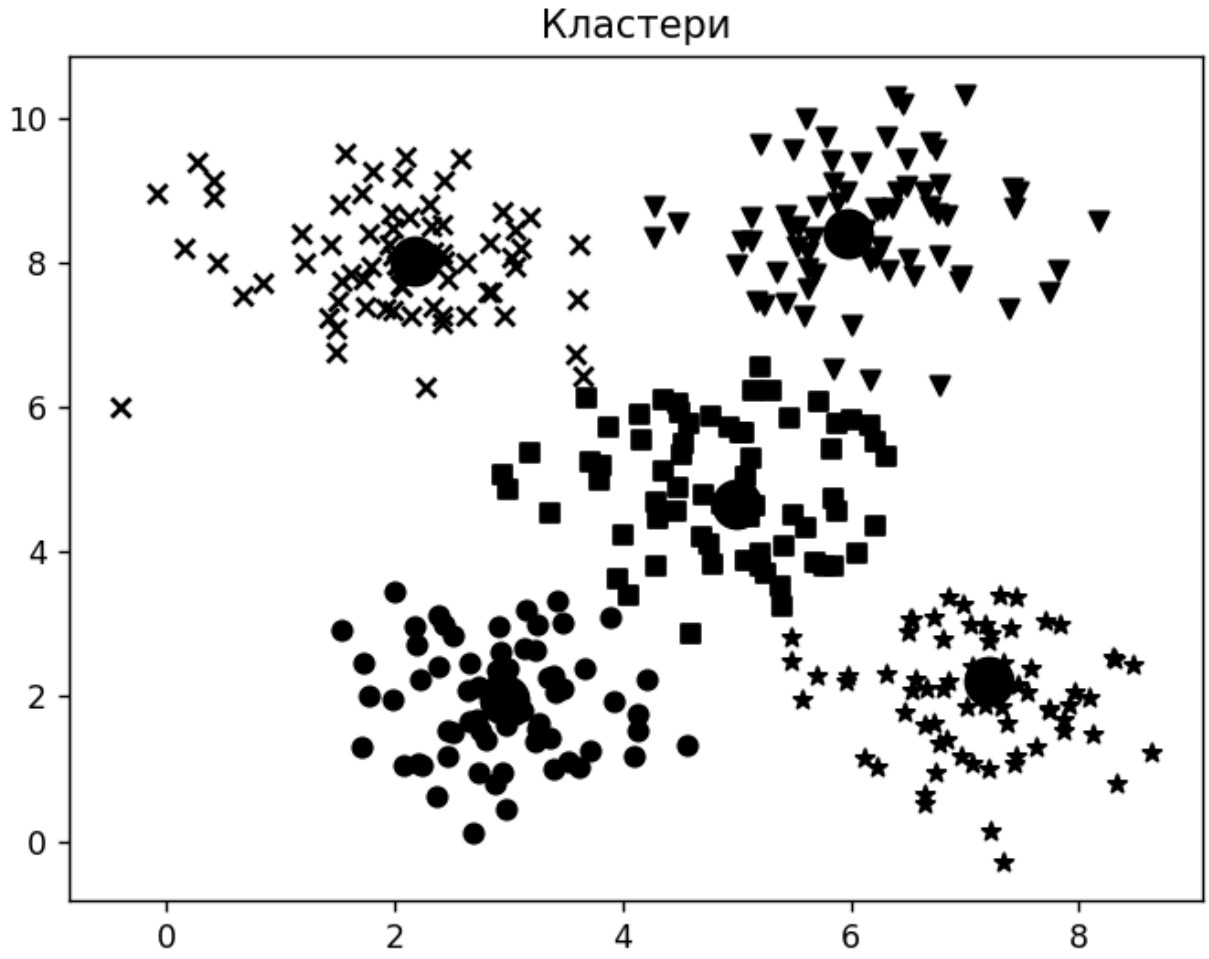
У цьому завданні було виконано кластеризацію класичного набору Iris, що містить дані про три типи ірисів. Алгоритм k-середніх з параметром k = 3 успішно розподілив вибірку на три підгрупи за ознаками довжини та ширини чашолистка. Отримані центри кластерів були автоматично знайдені. Хоча модель не використовувала реальні класи, візуалізація показала, що кластеризація доволі точно повторює структуру даних.

Завдання 2.3: Оцінка кількості кластерів з використанням методу зсуву середнього (Mean Shift).

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import MeanShift, estimate_bandwidth
from itertools import cycle

X = np.loadtxt('data_clustering.txt', delimiter=',')

bandwidth_X = estimate_bandwidth(X, quantile=0.1,
n_samples=len(X))

meanshift_model = MeanShift(bandwidth=bandwidth_X,
bin_seeding=True)
meanshift_model.fit(X)
```

```
cluster_centers = meanshift_model.cluster_centers_
print("\nCenters of clusters:\n", cluster_centers)

labels = meanshift_model.labels_
num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =",
num_clusters)

plt.figure()
markers = 'o*xvs^'
for i, marker in zip(range(num_clusters), markers):
    plt.scatter(X[labels == i, 0], X[labels == i, 1],
marker=marker, color='black')

for i in range(num_clusters):
    cluster_center = cluster_centers[i]
    plt.plot(cluster_center[0], cluster_center[1],
marker='o',
             markerfacecolor='black',
markeredgecolor='black',
             markersize=15)

plt.title('Кластери')
plt.show()
```

```
Centers of clusters:
 [[2.95568966 1.95775862]
 [7.20690909 2.20836364]
 [2.17603774 8.03283019]
 [5.97960784 8.39078431]
 [4.99466667 4.65844444]]
```



У ході виконання завдання було застосовано метод зсуву середнього (Mean Shift) для кластеризації двовимірних даних з файлу data_clustering.txt. Цей метод є непараметричним і не потребує попереднього задання кількості кластерів, що є його ключовою перевагою.

Після оцінки ширини вікна (параметра bandwidth), було виконано кластеризацію та побудовано графік, який демонструє розподіл точок за кластерами та центри кожного з них. Отримана кількість кластерів дозволяє зробити висновок про наявність груп у вибірці за критерієм щільності розміщення точок.

Алгоритм успішно виконав поставлене завдання — ідентифікував приховану кластерну структуру без попередніх припущень. Отримані результати підтверджують ефективність методу Mean Shift у задачах неконтрольованого навчання.

Завдання 2.4: Знаходження підгруп на фондовому ринку з використанням моделі поширення подібності

```
import datetime
import json
import numpy as np
import pandas as pd
from sklearn import covariance, cluster
```

```
input_file = 'company_symbol_mapping.json'

with open(input_file, 'r') as f:
    company_symbols_map = json.load(f)

symbols, names =
np.array(list(company_symbols_map.items())).T

df = pd.read_csv("stocks.csv")
df['date'] = pd.to_datetime(df['date'])

start_date = datetime.datetime(2003, 7, 3)
end_date = datetime.datetime(2007, 5, 4)
df = df[(df['date'] >= start_date) & (df['date'] <=
end_date)]

price_table = df.pivot(index='date', columns='symbol',
values='price')

price_table = price_table.dropna(axis=0)

quotes_diff = price_table.diff().dropna()

X = quotes_diff.values
X /= X.std(axis=0)

edge_model = covariance.GraphicalLassoCV()
edge_model.fit(X)

_, labels =
cluster.affinity_propagation(edge_model.covariance_)
num_labels = labels.max()

for i in range(num_labels + 1):
    print(f"Cluster {i + 1} =>", ', '.join(names[labels
== i]))
```

```
C:\Users\MikeIt\Desktop\ІАД\ПР_№1\venv\Scripts\python.exe C:\Users\MikeIt\Desktop\ІАД\ПР_№1\LR_4_task_4.py
Cluster 1 => Microsoft
Cluster 2 => Amazon
Cluster 3 => IBM, Apple, Google

Process finished with exit code 0
```

У межах четвертого завдання було реалізовано кластеризацію компаній на основі їхніх історичних фінансових показників за допомогою методів

машинного навчання. З використанням бібліотеки scikit-learn було побудовано граф залежностей між компаніями методом Graphical Lasso, а для кластеризації застосовано алгоритм Affinity Propagation.

У результаті аналізу було отримано три окремі кластери:

- Кластер 1 — Microsoft;
- Кластер 2 — Amazon;
- Кластер 3 — IBM, Apple, Google.

Отримані кластери вказують на наявність схожих ринкових трендів та поведінки акцій у межах кожної групи. Наприклад, компанії IBM, Apple і Google об’єдналися в один кластер, що свідчить про їхню подібну динаміку цін. У той же час Microsoft та Amazon продемонстрували унікальні поведінкові шаблони на ринку, що зумовило їх виділення в окремі кластери.

Таким чином, було успішно виконано кластеризацію фінансових даних, що дозволяє краще зрозуміти структуру взаємозв’язків між акціями провідних технологічних компаній.