

Лабораторна робота №7

Дослідження Методів Неконтрольованого Навчання

Виконав: ІПЗ-21-3 Осипчук Антон Олексійович

Github: https://github.com/AntonOsyphuk1/ai_lab/tree/main/lab7

Завдання 1. Кластеризація даних за допомогою методу k-середніх

Лістинг програми:

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

X = np.loadtxt('data/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('Input data')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())

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()])
```

```

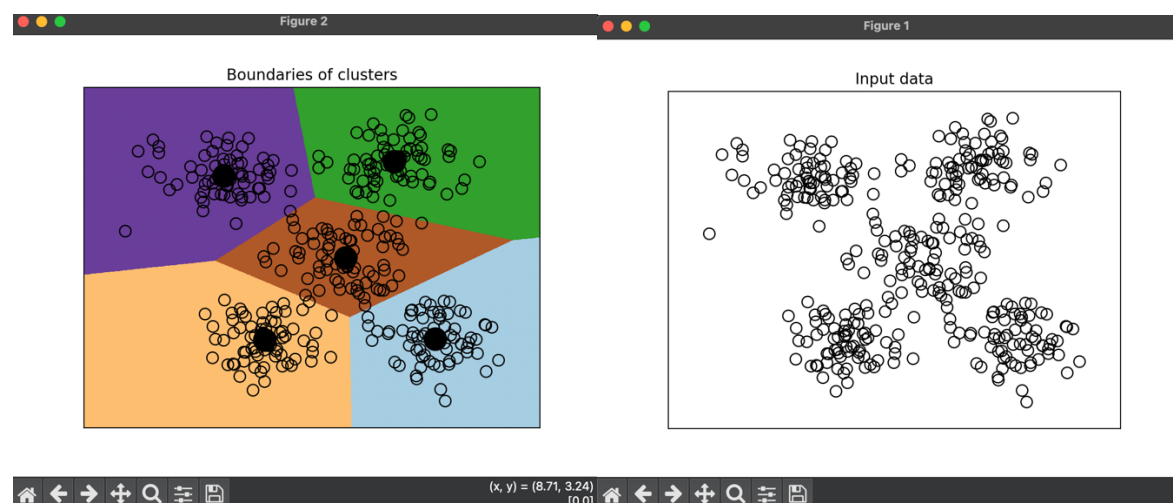
output = output.reshape(x_vals.shape)
plt.figure()
plt.clf()
plt.imshow(output, interpolation='nearest',
            extent=(x_vals.min(), x_vals.max(), y_vals.min(),
y_vals.max()),
            cmap=plt.cm.Paired,
            aspect='auto', origin='lower')

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('Boundaries of clusters')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()

```

Результат:



Завдання 2. Кластеризація K-середніх для набору даних Iris

Лістинг програми:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin
from sklearn.datasets import load_iris

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

# Використання KMeans для кластеризації
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

# Візуалізація результатів
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("KMeans Clustering with 5 Clusters")
plt.show()

# Визначення власної функції кластеризації
def find_clusters(X, n_clusters, rseed=2):
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]
    while True:
        labels = pairwise_distances_argmin(X, centers)
        new_centers = np.array([X[labels == i].mean(0) for i in
range(n_clusters)])
        if np.all(centers == new_centers):
            break
        centers = new_centers
    return centers, labels

# Використання функції для кластеризації
centers, labels = find_clusters(X, 3)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("Custom Clustering with 3 Clusters (rseed=2)")
plt.show()
```

```

centers, labels = find_clusters(X, 3, rseed=0)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("Custom Clustering with 3 Clusters (rseed=0)")
plt.show()

# Використання KMeans для кластеризації з іншими параметрами
labels = KMeans(3, random_state=0).fit_predict(X)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("KMeans Clustering with 3 Clusters")
plt.show()

```

Результат:



Завдання 3. Оцінка кількості кластерів з використанням методу зсуву середнього

Лістинг програми:

```

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

```

```

from itertools import cycle

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

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

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

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

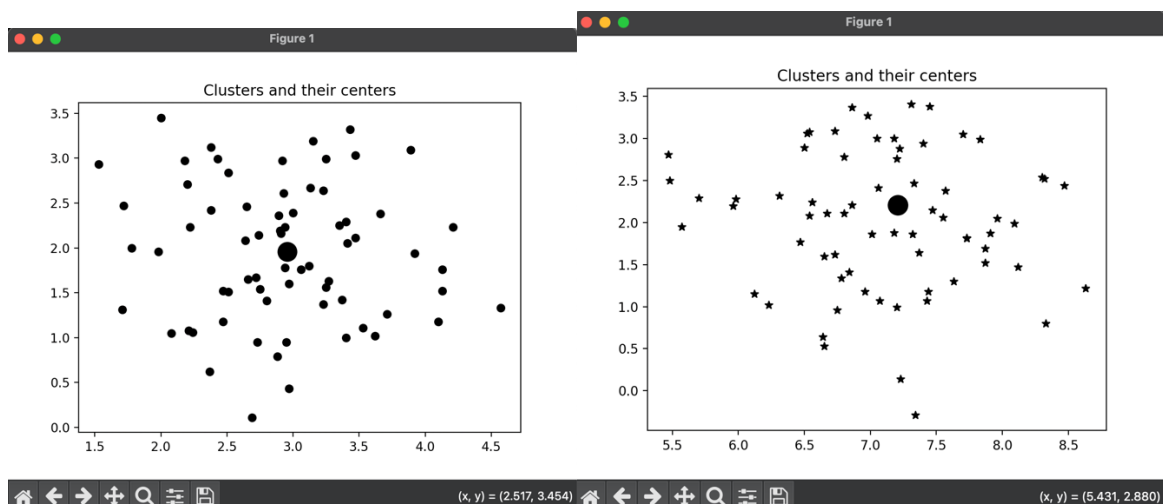
labels = meanshift_model.labels_
num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =", num_clusters)

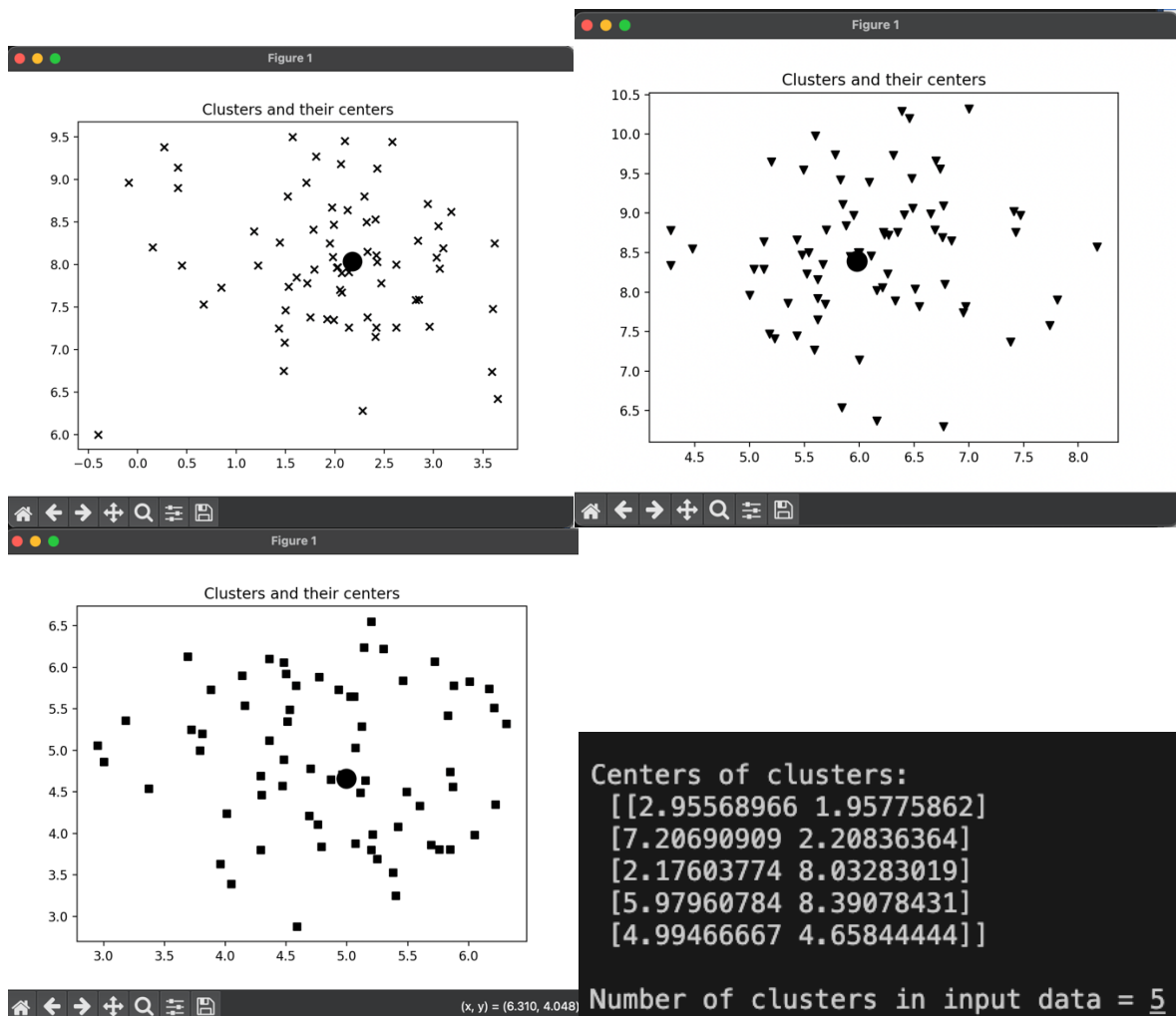
plt.figure()
markers = 'o*xvs'
for i, marker in zip(range(num_clusters), markers):
    plt.scatter(X[labels==i, 0], X[labels==i, 1], marker=marker,
color='black')

    cluster_center = cluster_centers[i]
    plt.plot(cluster_center[0], cluster_center[1], marker='o',
            markerfacecolor='black', markeredgecolor='black',
markersize=15)
plt.title('Clusters and their centers')
plt.show()

```

Результат:





Завдання 4. Знаходження підгруп на фондовому ринку з використанням моделі поширення подібності

Лістинг програми:

```

import datetime
import json
import numpy as np
import matplotlib.pyplot as plt
from sklearn import covariance, cluster
from matplotlib.finance import quotes_historical_yahoo_ochl as
quotes_yahoo

input_file = 'company_symbol_mapping.json'

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

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

start_date = datetime.datetime(2003, 7, 3)
  
```

```

end_date = datetime.datetime(2007, 5, 4)
quotes = [quotes_yahoo(symbol, start_date, end_date,
                        asobject=True) for symbol in symbols]

opening_quotes = np.array([quote.open for quote in
quotes]).astype(np.float)
closing_quotes = np.array([quote.close for quote in
quotes]).astype(np.float)

quotes_diff = closing_quotes - opening_quotes

X = quotes_diff.copy().T
X /= X.std(axis=0)

edge_model = covariance.GraphLassoCV()
with np.errstate(invalid='ignore'):
    edge_model.fit(X)

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

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

```

Результат:

```

⊗ (venv) antonosypchuk@MacBook-Air-Anton lab7 % python3 task4.py
Traceback (most recent call last):
  File "/Users/antonosypchuk/Desktop/lab/4 course/1 sem/AI/lab7/task4.py", line 6, in <module>
    from matplotlib.finance import quotes_historical_yahoo_ochl as quotes_yahoo
ModuleNotFoundError: No module named 'matplotlib.finance'

```