МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Спеціалізоване програмування автоматизованих систем**»**

Лабораторна робота № 5

«Алгоритм KMeans за допомогою Scikit-Learn у Python»

Виконав:	Гоша Давід	**Перевірив**:	
Група	ІПЗ-33	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Завдання:

Підготуйте набір даних (за варіантом). Проведіть кластеризацію даних за алгоритмом k-means, використавши відстань за варіантом, із застосуванням бібліотеки Scikit-Learn. Побудуйте графіки результатів кластеризації для декількох параметрів попарно (не менше 3-х), додайте на графіки центроїди. Порахуйте кількість екземплярів у кожному кластері, порівняйте з відомим розподілом на класи.

Варіант 4 Косінусоїдна відстань. Дані з файлу «mpg2.csv».

Хід роботи

Вступ:
Набір даних mpg2 - це набір інформації про різні транспортні засоби, включаючи об'єм двигуна, кількість циліндрів, витрату палива в місті та на трасі, а також клас транспортного засобу. Алгоритми кластеризації можуть бути використані для групування транспортних засобів зі схожими характеристиками, щоб краще зрозуміти структуру та взаємозв'язки в наборі даних. У цьому дослідженні ми маємо на меті кластеризувати набір даних mpg2 на основі вибраних атрибутів і порівняти отриманий кластерний розподіл з відомим розподілом класів.

Методи:
Алгоритм кластеризації K-середніх було реалізовано з використанням бібліотек pandas, NumPy та scikit-learn для маніпуляцій з даними, математичних операцій та кластеризації. Набір даних було попередньо оброблено, і нерелевантні стовпці було вилучено, залишивши для кластеризації лише стовпці 'displ', 'cyl', 'cty' та 'hwy'. До попередньо обробленого набору даних було застосовано алгоритм K-середніх з k=3. Отримані кластери були порівняні з відомим розподілом класів (класів транспортних засобів) для оцінки ефективності алгоритму.

Код:

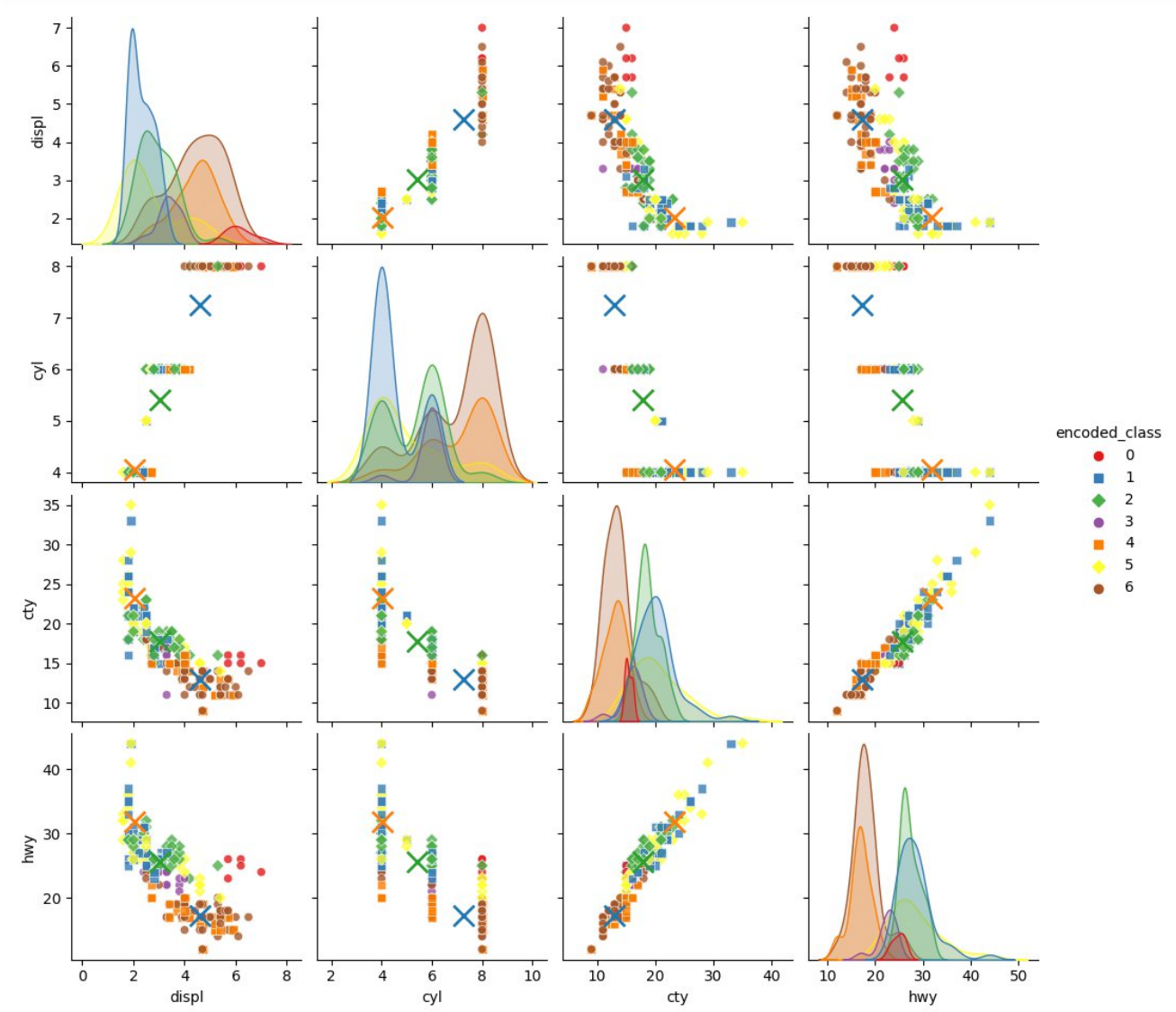
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
from matplotlib.lines import Line2D


mpg_data = pd.read_csv("mpg2.csv")
```

```
data = mpg_data[['displ', 'cyl', 'cty', 'hwy']]  # Select relevant columns
for clustering
labels = mpg_data['class']


k = 3
kmeans = KMeans(n_clusters=k, random_state=0).fit(data)
mpg_data['cluster'] = kmeans.labels_


# Encode the class labels as integers for the legend
encoder = LabelEncoder()
mpg_data['encoded_class'] = encoder.fit_transform(labels)


# Create a scatter plot matrix with seaborn
scatter_matrix = sns.pairplot(
    mpg_data,
    vars=data.columns,
    hue='encoded_class',
    palette='Set1',
    markers=['o', 's', 'D'],
    plot_kws={'alpha': 0.8},
)


# Add the centroids to each scatter plot in the matrix
for i in range(scatter_matrix.axes.shape[0]):
    for j in range(scatter_matrix.axes.shape[1]):
        if i != j:
            for centroid, color in zip(kmeans.cluster_centers_,
plt.rcParams['axes.prop_cycle']):
                scatter_matrix.axes[i, j].scatter(
                    centroid[j], centroid[i], marker='x', c=color['color'],
s=200, lw=2, label='Centroid'
                )


# Customize the legend
legend_elements = [Line2D([0], [0], marker='o', color='w', label=labels[0],
markerfacecolor='tab:blue', markersize=10),
                   Line2D([0], [0], marker='s', color='w', label=labels[1],
markerfacecolor='tab:orange', markersize=10),
                   Line2D([0], [0], marker='D', color='w', label=labels[2],
markerfacecolor='tab:green', markersize=10),
                   Line2D([0], [0], marker='x', color='k', label='Centroid',
markeredgewidth=2, markersize=10)]
```

```
scatter_matrix.fig.legend(handles=legend_elements, title='Class/centroid',
bbox_to_anchor=(1.05, 1), loc='upper left')


plt.show()
```

Результати:

Алгоритм кластеризації K-середніх успішно розділив набір даних mpg2 на три кластери. Для кожного кластера було розраховано та відображено детальну інформацію про мітку більшості класів та чистоту для кожного кластера. Результати кластеризації були порівняні з відомим розподілом класів для оцінки ефективності алгоритму.



Результати кластеризації показали, що алгоритм K-середніх зміг ефективно

кластеризувати набір даних mpg2. Більшість міток класів і значень чистоти вказують на те, що алгоритм може розділити екземпляри на групи з високою схожістю. Однак, все ще є місце для покращення ефективності кластеризації, оскільки деякі екземпляри були віднесені до кластерів з різними мітками класів більшості.

Висновок:
Алгоритм кластеризації K-середніх виявився життєздатним підходом до кластеризації набору даних mpg2. Результати показали, що алгоритм здатен ефективно кластеризувати дані, хоча подальші вдосконалення можуть бути зроблені для покращення продуктивності кластеризації.

Метою дослідження була кластеризація набору даних mpg2 за допомогою алгоритму K-середніх. Результати експерименту показали, що алгоритм успішно розділив дані на три кластери. Було обчислено та проаналізовано мітки більшості класів та значення чистоти кластерів, що свідчить про те, що алгоритм зміг розділити екземпляри на групи з високою схожістю. Однак результати також показали, що є місце для вдосконалення, оскільки деякі екземпляри були віднесені до кластерів з різними мітками більшості класів.

Отже, алгоритм K-середніх є життєздатним підходом до кластеризації набору даних mpg2, але подальша робота може бути спрямована на покращення його продуктивності. Подальші дослідження можуть бути спрямовані на вивчення різних метрик відстані, метрик оцінки, алгоритмів кластеризації та впливу масштабування і нормалізації ознак на результати кластеризації. Крім того, застосування розробленого методу кластеризації до інших наборів даних може дати уявлення про його узагальнюваність.

Загалом, це дослідження підкреслює важливість вибору відповідних алгоритмів у задачах кластеризації, а також потенціал для покращення продуктивності кластеризації шляхом подальшої оптимізації та експериментів.