CET 313 Artificial Intelligence
Workshop 8
Machine Learning Clustering

Aims of the workshop

In the lecture, we introduced the concept of unsupervised learning and clustering algorithms. Clustering is a fundamental technique in machine learning and data analysis that involves grouping similar data points together to uncover underlying patterns and structures within datasets.

In this tutorial, we will explore the implementation of various clustering algorithms using the Python programming language. Starting with an introduction to clustering concepts, we will delve into practical exercises with hands-on implementations of popular algorithms such as K-Means.

By the end of this tutorial, you will have a foundational understanding of how to apply clustering algorithms to a real-world dataset and evaluate the performance of the model using different metrics.

Let's begin the tutorial by importing the necessary libraries and loading the dataset.

Feel free to discuss your work with peers, or with any member of the teaching staff.

Reminder

We encourage you to discuss the content of the workshop with the delivery team and any findings you gather from the session.

Workshops are not isolated, if you have questions from previous weeks, or lecture content, please come and talk to us.

Exercises herein represent an example of what to do; feel free to expand upon this.

Helpful Resources

Scikit-learn clustering
2.3. Clustering — scikit-learn 1.3.2 documentation

Book chapter
https://learning.oreilly.com/library/view/hands-on-machine-learning/9781098125967/ch09.html

Dataset Hosting Websites:

1. Kaggle:

Kaggle is an online community of data scientists and machine learning practitioners. It offers a platform for data science competitions, hosting datasets and code, and providing a range of educational resources. Kaggle is a valuable resource for data science enthusiasts looking to hone their skills, collaborate with others, and work on real-world data science problems.

https://www.kaggle.com/

2. UCI Machine Learning Repository:

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators used by the machine learning community for the empirical analysis of machine learning algorithms. It is a widely used repository that provides various datasets for experimental research on machine learning techniques.

https://archive.ics.uci.edu/

3. UCI KDD Archive:

The UCI KDD Archive is a subsection of the UCI Machine Learning Repository, specifically focusing on datasets that are relevant to the KDD (Knowledge Discovery in Databases) process. It includes a variety of datasets that can be used for tasks such as classification, clustering, regression, and other machine learning tasks in the context of data mining and knowledge discovery.

http://kdd.ics.uci.edu/summary.data.application.html

4. UK government, NHS digital, etc.

Environment setup:

In this workshop we will be using few libraries, for the machine learning, thus initially we need to install them. Initial you need to start a new notebook and type the installation commands in code cells, once the installation is complete you need to restart the kernel. **[If you have done these last week, you do not need to re install]**

1. Install Pandas

```
!pip install pandas
```

2. Install NumPy

```
!pip install numpy
```

3. Install Matplotlib

```
!pip install matplotlib
```

4. Install scikit-learn

```
!pip install scikit-learn
```

Exercises

You may find it useful to keep track of your answers from workshops in a separate document, especially for any research tasks.
Where questions are asked of you, this is intended to make you think; it would be wise to write down your responses formally.

Exercise 1: Answer the quiz available via canvas.
Exercise 2: Install the required libraries and its dependences.

If you are using the university PC you will need to update a library using the following script:

```
!pip install --upgrade threadpoolctl
```

Exercise 3: Initially we will start with a simple k-means algorithm implantation.

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

random_state_value = 42
# Generate random data for clustering
np.random.seed(random_state_value)
data, _ = make_blobs(n_samples=300, centers=3, cluster_std=1.0,
random_state=42)

# Function to perform K-means clustering and plot the results
def plot_kmeans_clusters(data, n_clusters):
    kmeans = KMeans(n_clusters=n_clusters, random_state=random_state_value)
    labels = kmeans.fit_predict(data)

    plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis',
edgecolor='k', s=50)
    plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
c='red', marker='x', s=200, label='Centroids')
    plt.title(f'K-means Clustering with {n_clusters} Clusters')
    plt.legend()
    plt.show()

plot_kmeans_clusters(data, n_clusters=2)
```

Exercise 4: Read the script and understand how it works.
Exercise 5: Update the script to cluster the data with different cluster number ranging from 2 clusters to 10 (using loop).
Exercise 6: Experiment with different random state value and comment on its effect on the algorithm.

Exercise 7: Now we will use the iris dataset and use the k means algorithm to cluster it

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

random_state_value = 42

# Load the Iris dataset
iris = load_iris()
data = iris.data

# Function to perform K-means clustering and plot the results
def plot_kmeans_clusters(data, n_clusters):
    kmeans = KMeans(n_clusters=n_clusters, random_state=random_state_value)
    labels = kmeans.fit_predict(data)

    plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis',
edgecolor='k', s=50)
    plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
c='red', marker='x', s=200, label='Centroids')
    plt.title(f'K-means Clustering with {n_clusters} Clusters')
    plt.legend()
    plt.show()

# Plot the data with different numbers of clusters
N = [2, 3, 4, 5, 6, 7]
for n in N:
    plot_kmeans_clusters(data, n_clusters=n)
```

Exercise 8: Now you can check the effect of basic assumption in l-means algorithm by running the codes from this tutorial:

Demonstration of k-means assumptions — scikit-learn 1.3.2 documentation

Exercise 9: [optional] Complete this tutorial to get a good understanding of the clustering algorithms on real-word dataset:
Online Retail K-Means & Hierarchical Clustering | Kaggle

Make sure you upload your notebook with the solutions to your eportfoilio

END OF EXERCISES