K-MEANS CLUSTERING ALGORITHM IN MACHINE LEARNING

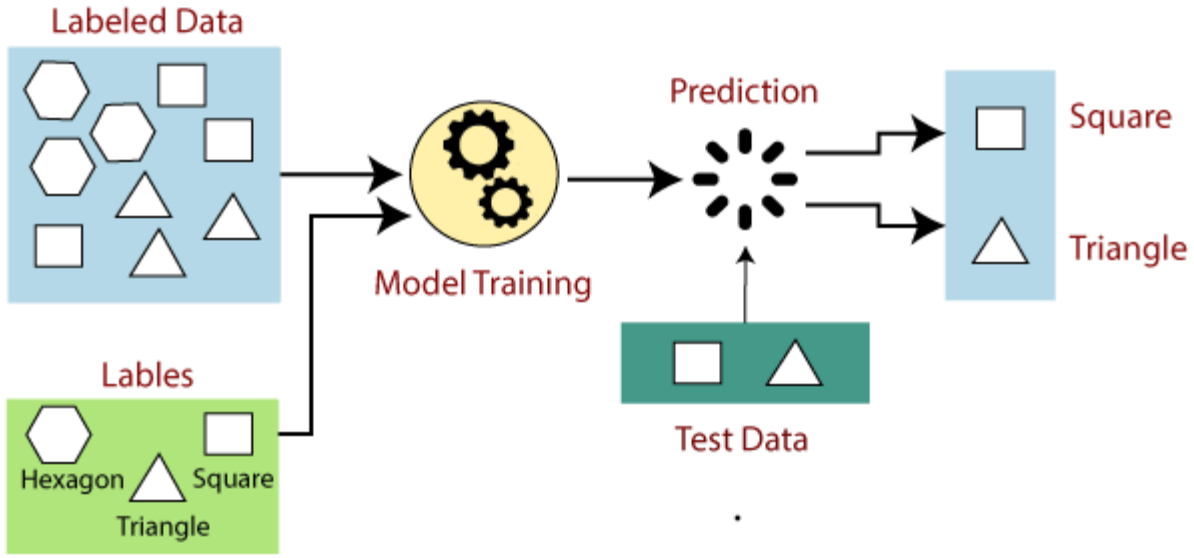
Ashkan Damavandi

Sharif University of Technology
Department of Physics
Spring 2025

LABELED DATA AND UNLABELED DATA

Data Labeling

- Data labeling, or data annotation, is part of the preprocessing stage when developing a machine learning (ML) model.
- Data labeling requires the identification of raw data (like images, text files, videos) and then the addition of one or more labels to that data to specify its context for the models, allowing the machine learning model to make accurate predictions.
- Companies integrate software, processes and data annotators to clean, structure and label data. This training data becomes the foundation for machine learning models. These labels allow analysts to isolate variables within datasets, and this, in turn, enables the selection of optimal data predictors for ML models. The labels identify the appropriate data vectors to be pulled in for model training, where the model, then, learns to make the best predictions.



Example of data labeling process in Machine Learning [5]

LABELED DATA AND UNLABELED DATA

Labeled data

- Labeled data is used in supervised learning
- Labeled data is more difficult to acquire and store and is time consuming and expensive
- Labeled data can be used to determine actionable insights

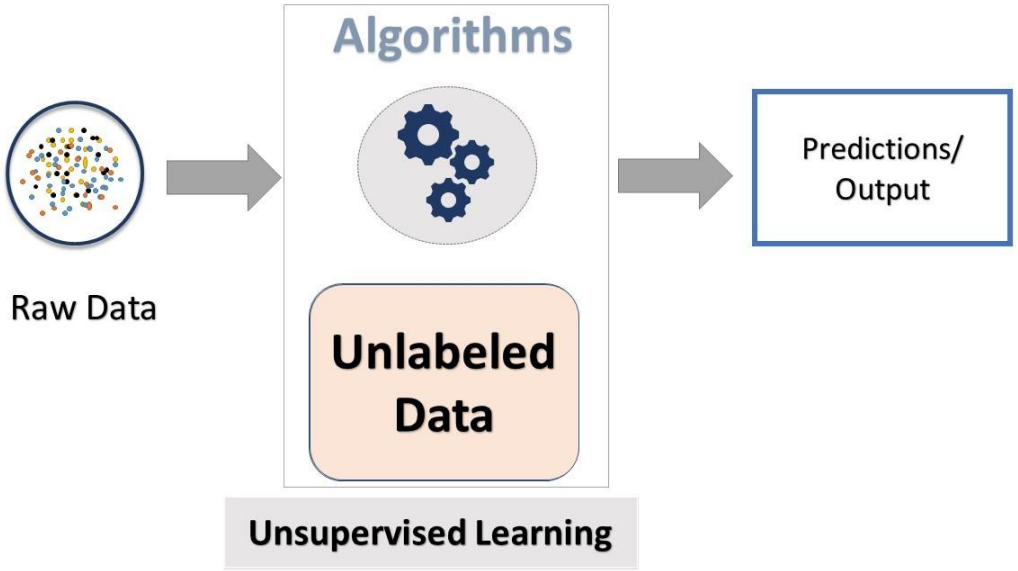
Unlabeled data

- Unlabeled data is used in unsupervised learning
- Unlabeled data is easier to acquire and store
- Unlabeled data is more limited in its usefulness

K-MEANS CLUSTERING ALGORITHM

Unsupervised Machine Learning

- Unsupervised learning methods can help discover new clusters of data, allowing for new categorizations when labeling
- Unsupervised learning is the type of Machine Learning, which is used to find hidden data patterns, or it is often used when you have limited understanding of the data and want to explore the similarities. Unlike Supervised Learning, Unsupervised Learning does not consist predefined features or labelled datasets and does not give the predictions or trained models directly.

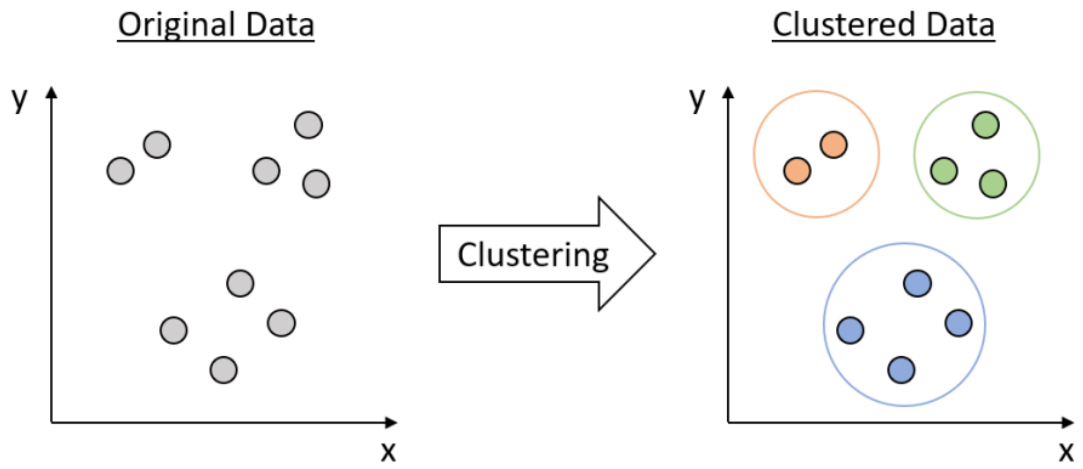


Unsupervised Machine Learning Working [6]

K-MEANS CLUSTERING ALGORITHM

Clustering in Machine Learning

- Clustering is an unsupervised learning technique used to find the subgroups (also known as Clusters) in a dataset.
- Clustering is a technique in which similar data points in dataset will be grouped based on certain criteria such as their shapes, size, taste or color.
- The clustering technique is prevalent in many fields and many algorithms exist to perform it.

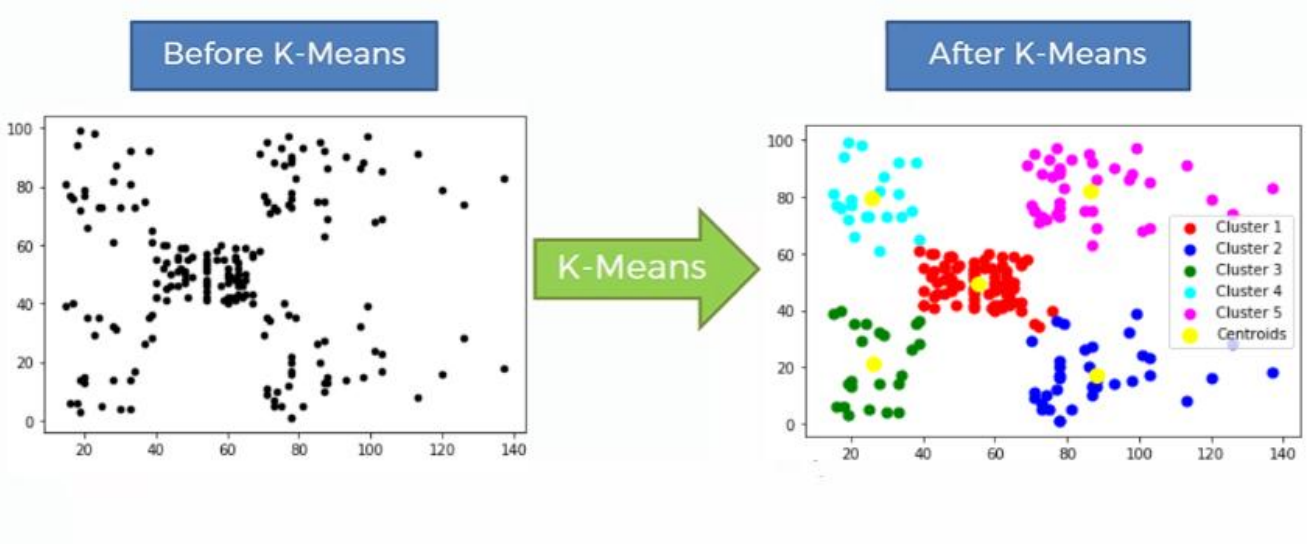


Clustering Process [7]

K-MEANS CLUSTERING ALGORITHM

K-means meaning

- K-Means is one of the most popular clustering algorithms and is a simple and elegant approach to partitioning data samples into pre-defined “K” distinct and non-overlapping clusters.
- K-means stores K centroids that it uses to define clusters. The value of “K” in the K-means algorithm depends on the user’s choice.
- A point is considered to be in a particular cluster if it is closer to that cluster's centroid (points which are the center of a cluster) than any other centroid.
- K-Means finds the best centroids by alternating between assigning data points to clusters based on the current centroids and choosing centroids based on the current assignment of data points to clusters.
- In the K-means algorithm, every data sample from the dataset will follow two fundamental properties: First, each data sample belongs to at least one of the ‘K’ clusters. In simple terms, there can’t be any sample that is not a part of any of the clusters. Second, no data sample will belong to more than one cluster. In simple terms, one sample cannot be present in two (or more) clusters at the same time.



Visual example of K-means clustering algorithm [3]

K-MEANS CLUSTERING ALGORITHM

A simple pseudocode for the K-means algorithm

Input:
- *Dataset X with n data points*
- *Number of clusters K*

Output:
- *K cluster assignments*
- *K centroids*

Steps:

1. Initialize K centroids randomly from the data points

2. Repeat until convergence (centroids don't change significantly):

a. Assignment Step:

For each data point in X:
- *Calculate distance from the point to each centroid*
- *Assign the point to the nearest centroid (i.e., cluster)*

b. Update Step:

For each cluster:
- *Recalculate the centroid as the mean of all points assigned to that cluster*

3. Return:
- *Final cluster assignments for each data point*
- *Final positions of the centroids*

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }

The K-means clustering algorithm [1]

K-MEANS CLUSTERING ALGORITHM

Steps of K-means clustering algorithm

1) Defining the number of clusters:

- Suppose we have some data samples (X1, X2, …, Xn), and we want to divide these data samples into “K” clusters.
- Accept data inputs and pre-define the number of groups we want to form. In our case, we have “n” data samples, and we want to divide these samples into K clusters.

2) Initialization of K means:

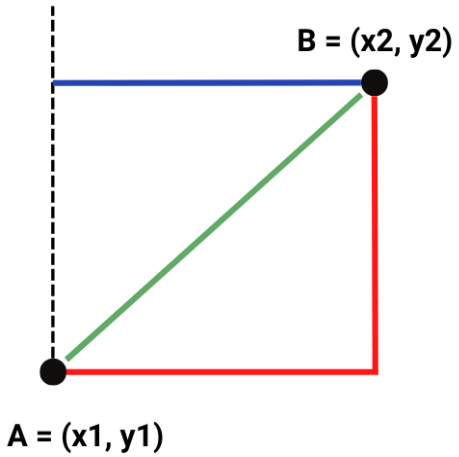
- Initialize the first K means for K clusters.
- There can be two ways to do this: Pick the first K samples, or randomly pick K elements from all the available data samples. These K samples will be treated as temporary centroids (also called the mean).

3) Assigning data points to any of the K clusters:

- Now, we have (n - k) data samples left. For each sample, we need to calculate its distance from all the K centroids, and later it will be assigned to only one of the K clusters for which distance would be minimum.
- If the distances from the two clusters are equal and minimum, we can pick any one cluster and put that sample in that.

K-MEANS CLUSTERING ALGORITHM

Different ways of measuring distance between two data points



Euclidean Distance

$$||A - B|| = ((x1-x2)^2 + (y1-y2)^2)^{1/2}$$

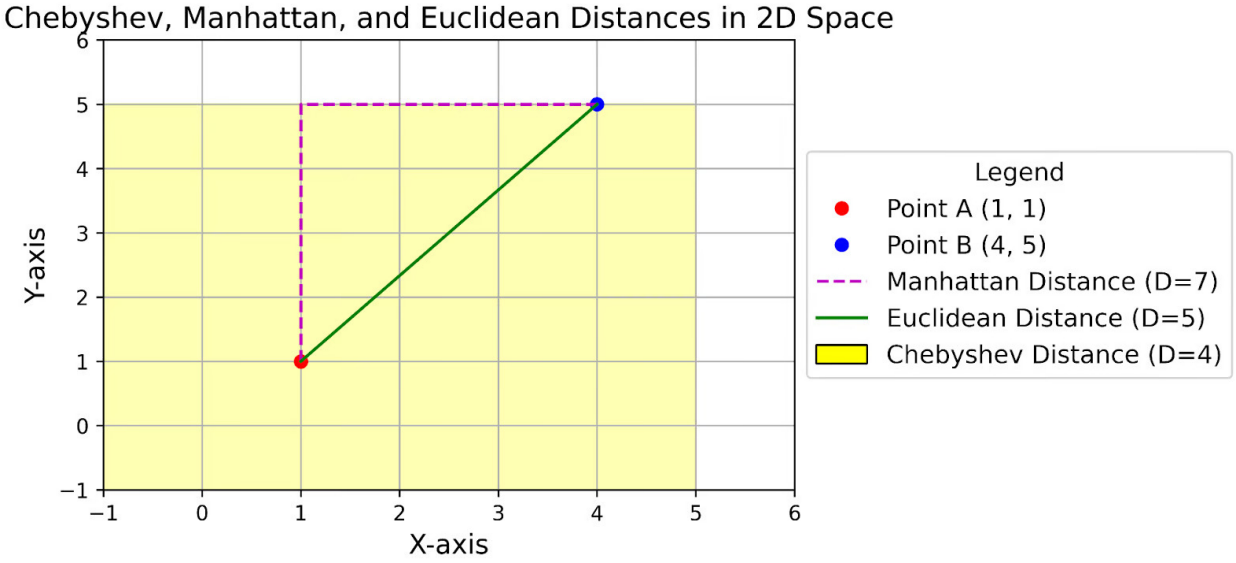
Manhattan Distance

$$||A - B|| = |x1-x2| + |y1-y2|$$

Chebyshev Distance

$$||A - B|| = \max\{(x1-x2), (y1-y2)\}$$

There are different ways in which we can measure that distance. [2]



Explanation of different ways of distance measurement. [8]

K-MEANS CLUSTERING ALGORITHM

Steps of K-means clustering algorithm

4) Updating centroids:

- After all the data points have been assigned to a cluster, we need to calculate the new centroids for every cluster.
- This time the centroid calculation will not be random.
- The new centroid is calculated as the mean of all the data points assigned to that cluster.
- At this step, we will have K centroids that can differ from the earlier chosen random K samples.
- There can be a possibility that no other data sample was assigned to one cluster. In that case, the centroid for that cluster will not update.

5) Repeat steps 3 and 4:

- We will repeat steps 3 and 4 until the entire n samples converge.
- Convergence means that there will be no or negligible movement of samples among clusters.
- In this way, K-means groups the data samples into the desired number of clusters.
- The steps are straightforward and intuitive, one of the most important reasons for the algorithm's popularity.

K-MEANS CLUSTERING ALGORITHM

My code (Doesn’t work perfectly)

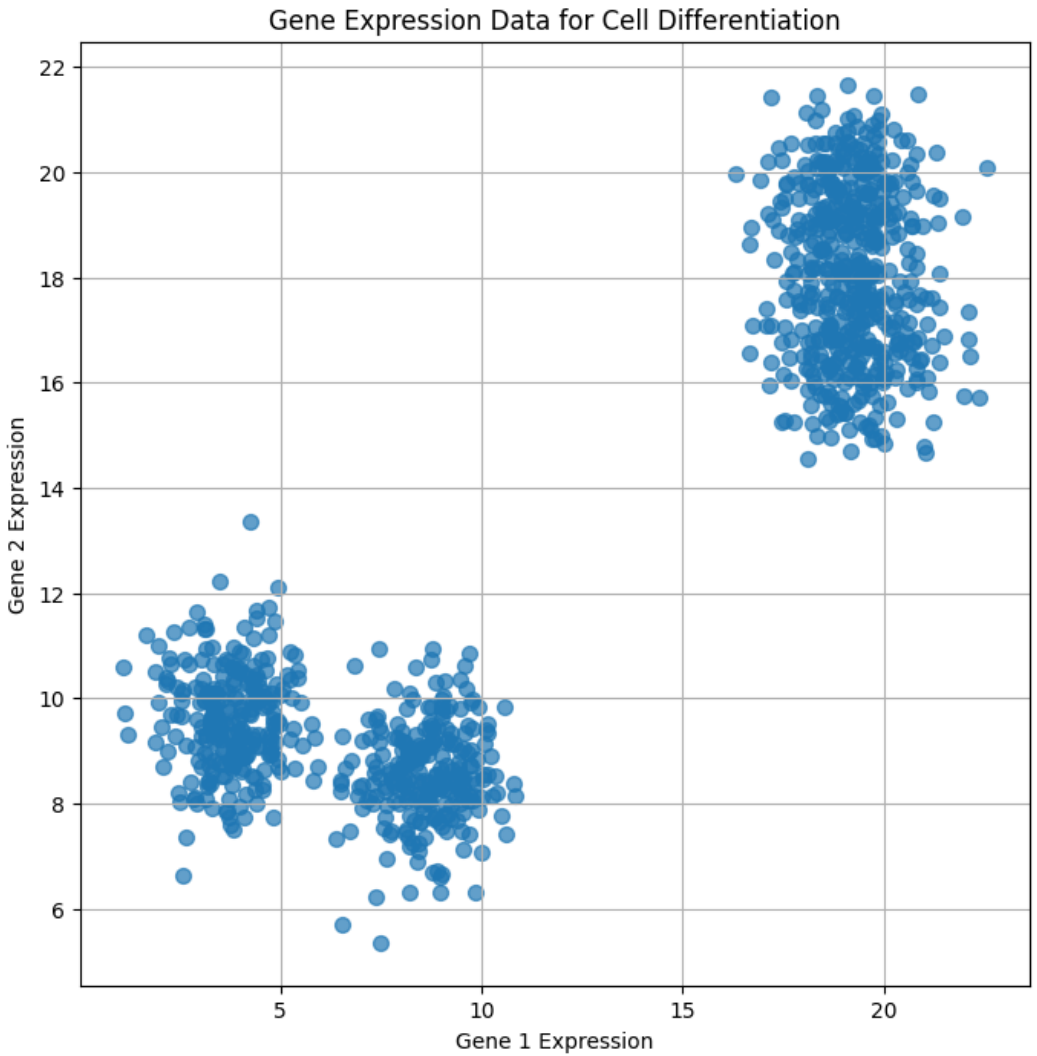
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


def cell_differentiation_data(num_samples=100, num_genes=10):
    np.random.seed(42)
    cell_types = ['Type A', 'Type B', 'Type C', 'Type D']
    num_cell_types = len(cell_types)
    data = []

    for i in range(num_cell_types):
        means = (np.random.rand(num_genes) * 10) + (list(cell_types).index(cell_types[i]) * 5)
        for j in range(num_samples // num_cell_types):
            data.append(np.random.normal(means, 1))

    gene_expression_df = pd.DataFrame(data)
    column_names = []
    for k in range(num_genes):
        column_names.append(f'Gene {k + 1}')

    gene_expression_df.columns = column_names
    samples_per_type = num_samples // num_cell_types
    cell_type_list = []

    for l in range(num_cell_types):
        for m in range(samples_per_type):
            cell_type_list.append(cell_types[l])

    cell_type_array = np.array(cell_type_list)

    return gene_expression_df, cell_type_array


def plot_data(data, title='Gene Expression Data for Cell Differentiation'):
    plt.figure(figsize=(8, 8))
    plt.scatter(data.iloc[:, 0], data.iloc[:, 1], s=50, alpha=0.7)
    plt.title(title)
    plt.xlabel('Gene 1 Expression')
    plt.ylabel('Gene 2 Expression')
    plt.grid()
    plt.show()
```

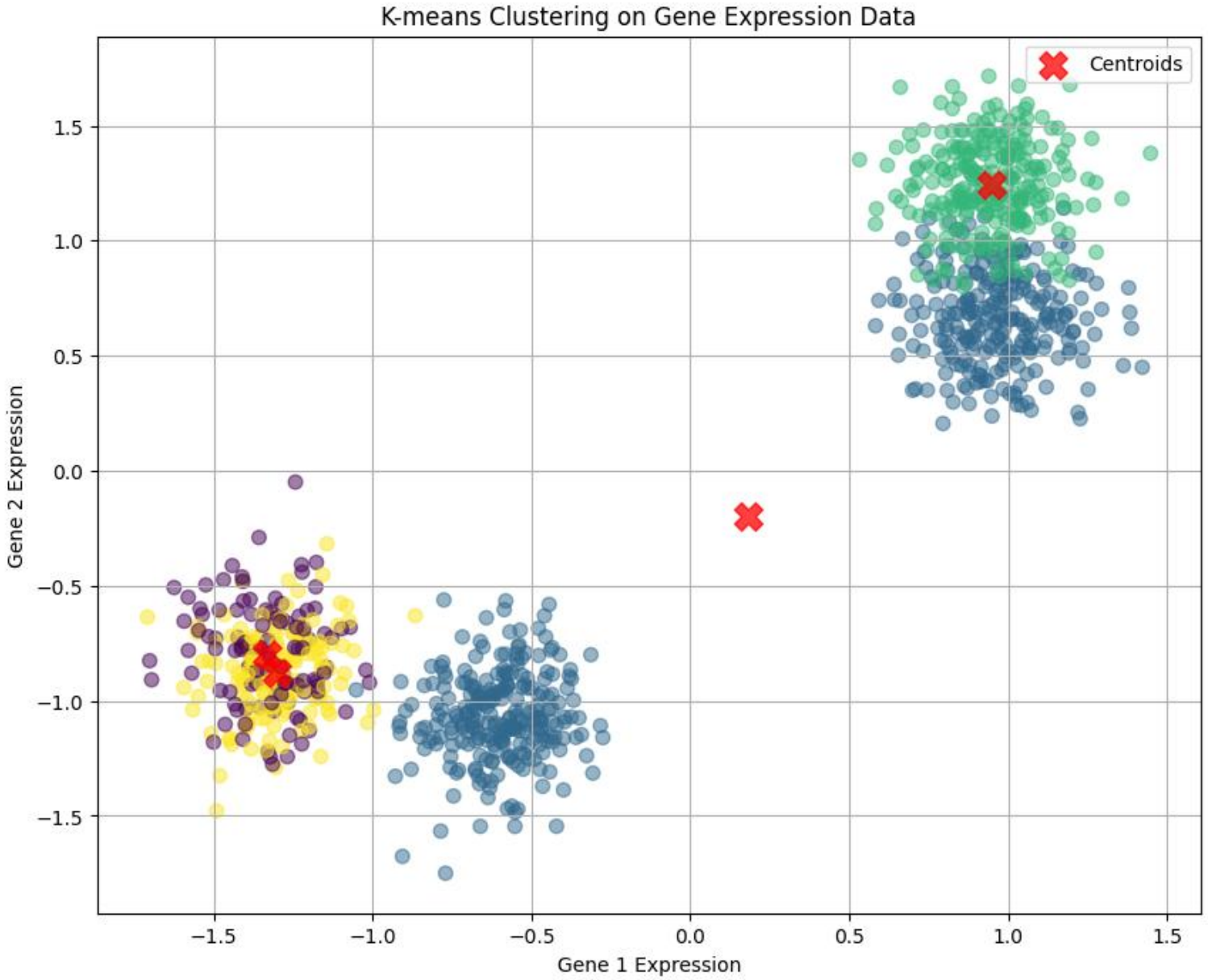
```python
def standardize_data(data):
    mean = np.mean(data, axis=0)
    std = np.std(data, axis=0)

    return (data - mean) / std


def initialize_centroids(data, k):
    m, n = data.shape
    indices = np.random.choice(m, k, replace=False)

    return data.iloc[indices].values


def assign_clusters(data, centroids):
    distances = np.zeros((data.shape[0], centroids.shape[0]))

    for i in range(centroids.shape[0]):
        distances[:, i] = np.linalg.norm(data.values - centroids[i], axis=1)

    return np.argmin(distances, axis=1)


def update_centroids(data, clusters, k):
    centroids = np.zeros((k, data.shape[1]))

    for i in range(k):
        centroids[i] = data.values[clusters == i].mean(axis=0)

    return centroids


def kmeans_algorithm(data, k, max_iterations):
    centroids = initialize_centroids(data, k)

    for i in range(max_iterations):
        clusters = assign_clusters(data, centroids)
        new_centroids = update_centroids(data, clusters, k)
        if np.all(centroids == new_centroids):  # Check for convergence
            break
        centroids = new_centroids

    return clusters, centroids
```

```python
def plot_clusters(data, clusters, centroids):
    plt.figure(figsize=(10, 8))
    plt.scatter(data.iloc[:, 0], data.iloc[:, 1], c=clusters, s=50, cmap='viridis', alpha=0.5)
    plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, alpha=0.75, marker='X', label='Centroids')
    plt.title('K-means Clustering on Gene Expression Data')
    plt.xlabel('Gene 1 Expression')
    plt.ylabel('Gene 2 Expression')
    plt.legend()
    plt.grid()
    plt.show()


num_samples = 1000
num_clusters = 4
max_iterations = 100
data, cell_types = cell_differentiation_data(num_samples=num_samples)
plot_data(data)
standardized_data = standardize_data(data)
clusters, centroids = kmeans_algorithm(standardized_data, num_clusters, max_iterations)
plot_clusters(standardized_data, clusters, centroids)
```

K-MEANS CLUSTERING ALGORITHM

My code's results (Doesn't work perfectly)





K-MEANS CLUSTERING ALGORITHM

Limitations of K-means clustering algorithm

1. K-means algorithm is computationally expensive. It requires significant computational resources and time to process large amounts of data. The time required is proportional to the number of data items, the number of clusters (k) and the number of iterations.
2. The quality of the resulting clusters highly depends on the initial selection of K clusters. The algorithm randomly picks K clusters in the first step, which can lead to suboptimal results if the initial clusters are not chosen carefully.
3. There can be situations where the K-means algorithm suffers from an empty cluster problem. This can happen when the initial K centroids are selected randomly, and one of them happens to be an outlier. In such cases, the cluster associated with that centroid might become empty.

REFERENCES

[1]: *https://stanford.edu/*

[2]: *https://www.enjoyalgorithms.com/*

[3]: *https://nouralserw.medium.com/k-means-the-maths-behind-it-how-it-works-and-an-example-67fdcfcb80f0*

[4]: *https://www.ibm.com/*

[5]: *https://www.javatpoint.com/supervised-machine-learning*

[6]: *https://www.edushots.com/Machine-Learning/unsupervised-machine-learning-overview*

[7]: *https://www.linkedin.com/pulse/what-clustering-machine-learning-avishek-patra-ap/*

[8]: *https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.datacamp.com%2Ftutorial%2Fchebyshev-distance&psig=AOvVaw2VRXZnJDBgSA4fpW572ICD&ust=1744304402548000&source=images&cd=vfe&opi=89978449&ved=0CBYQjhxqGAoTCIi-prW2y4wDFQAAAAAdAAAAABCIAg*