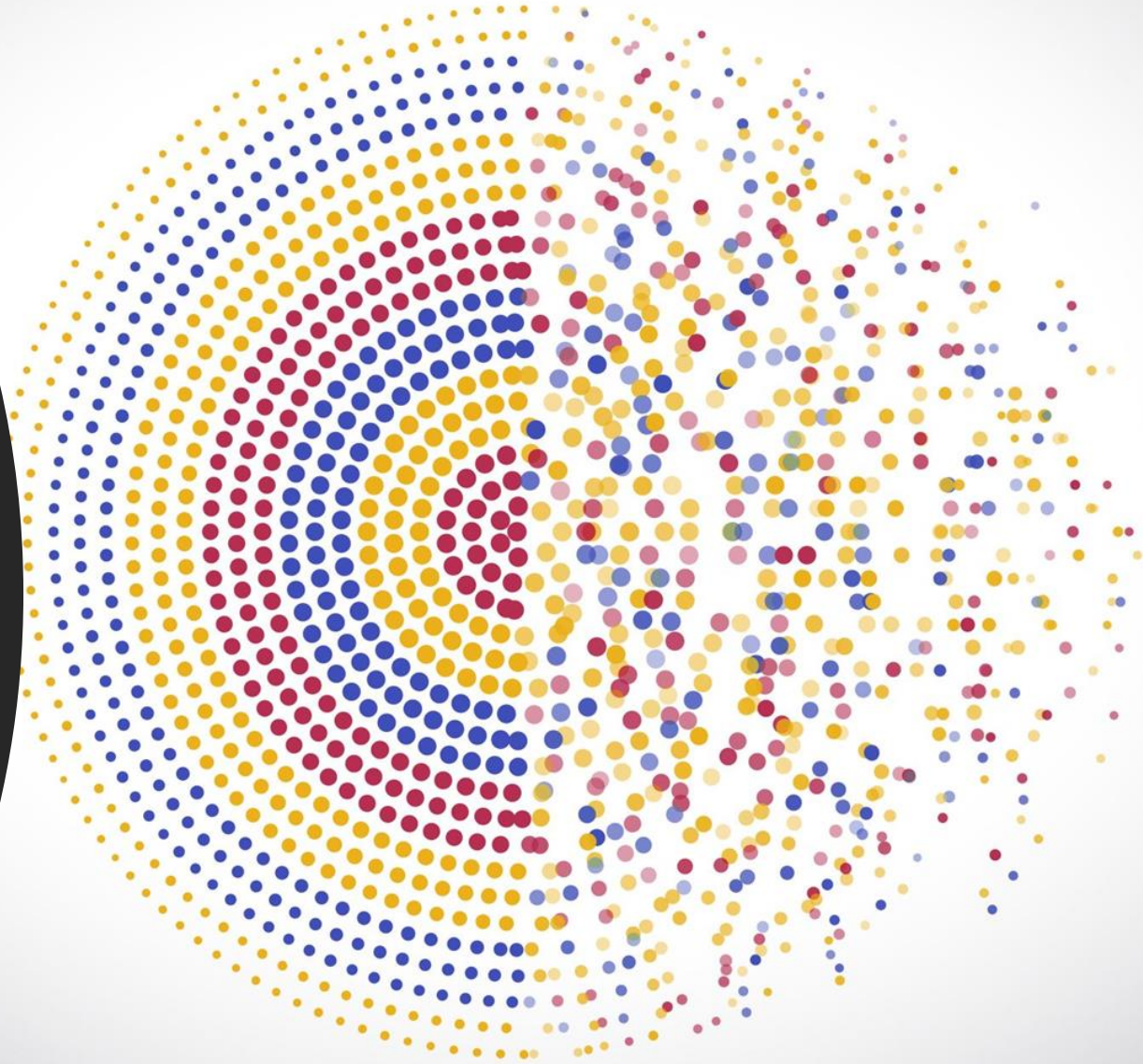
Similarity Measure

Dr. Mohamed AlHajri



Similarity Measure

- In many machine learning and data mining tasks, understanding the relationships between data points is essential. Various techniques measure these relationships, each with unique applications and mathematical properties.
- we will explore the major techniques used to measure similarity, delve into the mathematics behind them, provide examples using real-world datasets, and discuss their limitations and applications.
    - Pearson correlation
    - Euclidean distance
    - Manhattan distance
    - Cosine similarity
    - Jaccard similarity

Pearson Correlation

- The Pearson correlation coefficient measures the **linear relationship** between two variables. It is suitable for detecting linear correlations but fails with non-linear relationships.

$$\rho(A,B) = \frac{cov(A,B)}{\sigma_A \sigma_B}$$

- **Applications:**
  - *Feature selection in machine learning.*
  - *Linear regression for analyzing correlations between variables.*

Pearson Correlation - Linear

$$\rho(A,B) = \frac{cov(A,B)}{\sigma_A \sigma_B}$$

Numerical Exercise (Linear Case): Given $A = [1,2,3,4,5]$ and $B = [2,4,6,8,10]$

1.Mean:

$$\bar{A} = 3, \qquad \bar{B} = 6$$

2. Covariance:

$$Cov(A,B) = \frac{1}{5}\sum(A_i - \bar{A})(B_i - \bar{B}) = 4$$

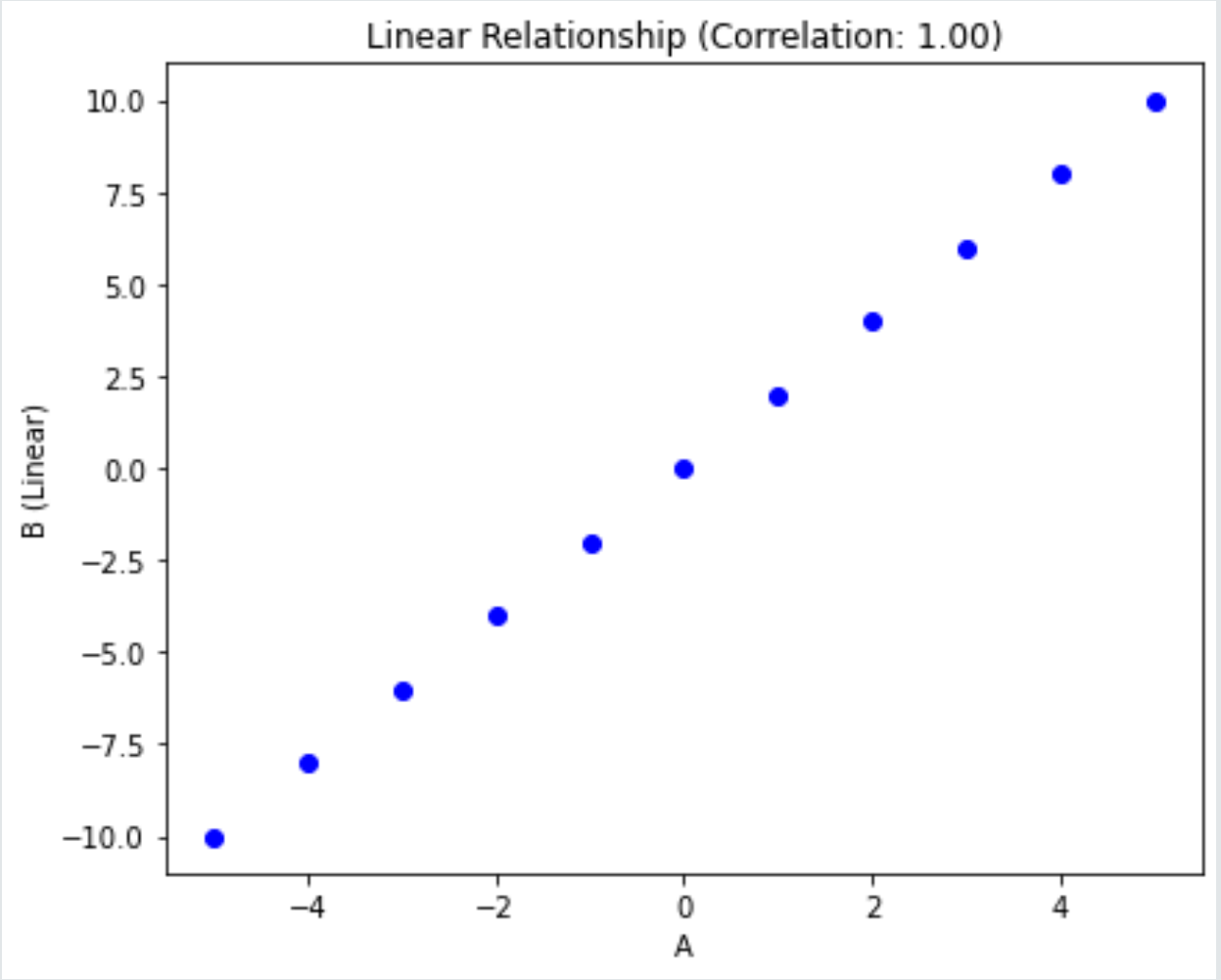
3. Standard Deviations:

$$\sigma_A = \sqrt{2}, \sigma_B = \sqrt{8}$$

4. Pearson Correlation:

$$\rho(A,B) = \frac{4}{\sqrt{2} * \sqrt{8}} = 1$$

Pearson Correlation - Linear



Pearson Correlation – Non linear

$$\rho(A,B) = \frac{cov(A,B)}{\sigma_A \sigma_B}$$

Numerical Exercise (Non-Linear Case): Given $A = [-5,-4,-3,-2,-1,0,1,2,3,4,5]$ and $B = [25,16,9,4,1,0,1,4,9,16,25]$

1.Mean:

$$\bar{A} = 0\,, \qquad \bar{B} = 10$$

2. Covariance:

$$Cov(A,B) = \frac{1}{10}\sum(A_i - \bar{A})(B_i - \bar{B}) = 0$$

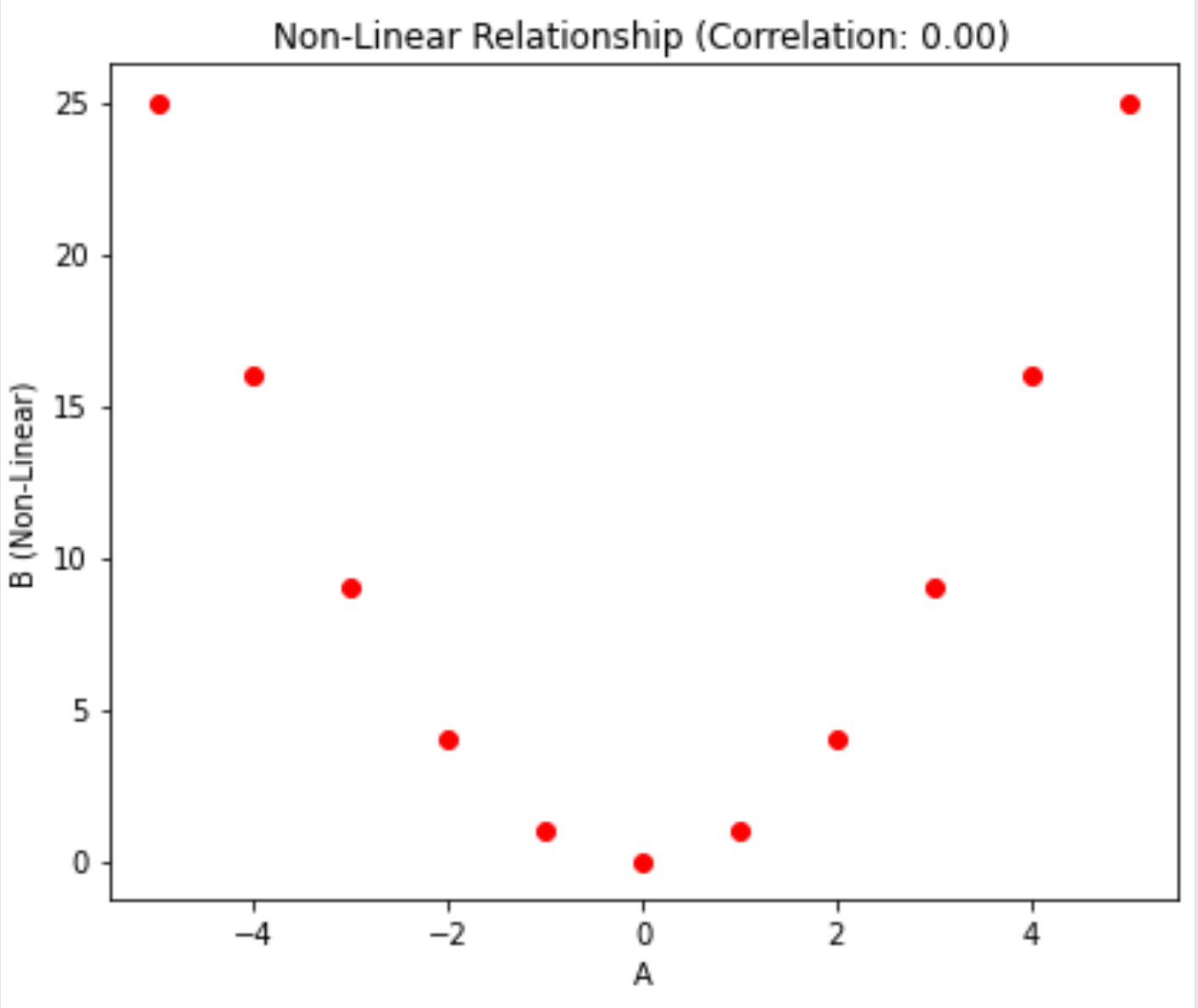
3. Standard Deviations:

$$\sigma_A =, \sigma_B =$$

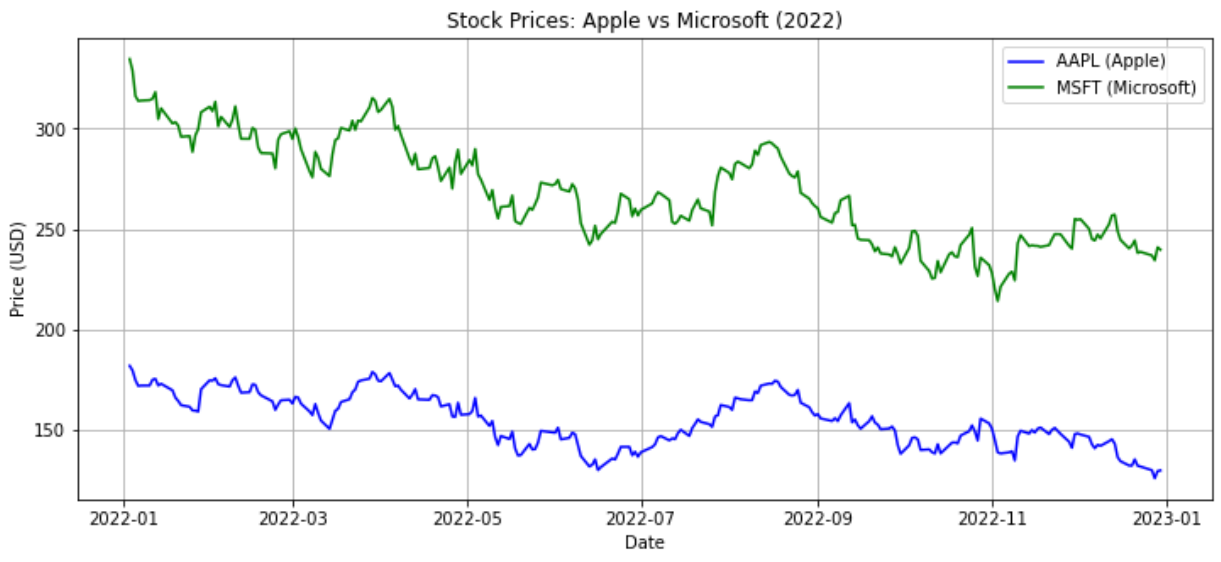
4. Pearson Correlation:

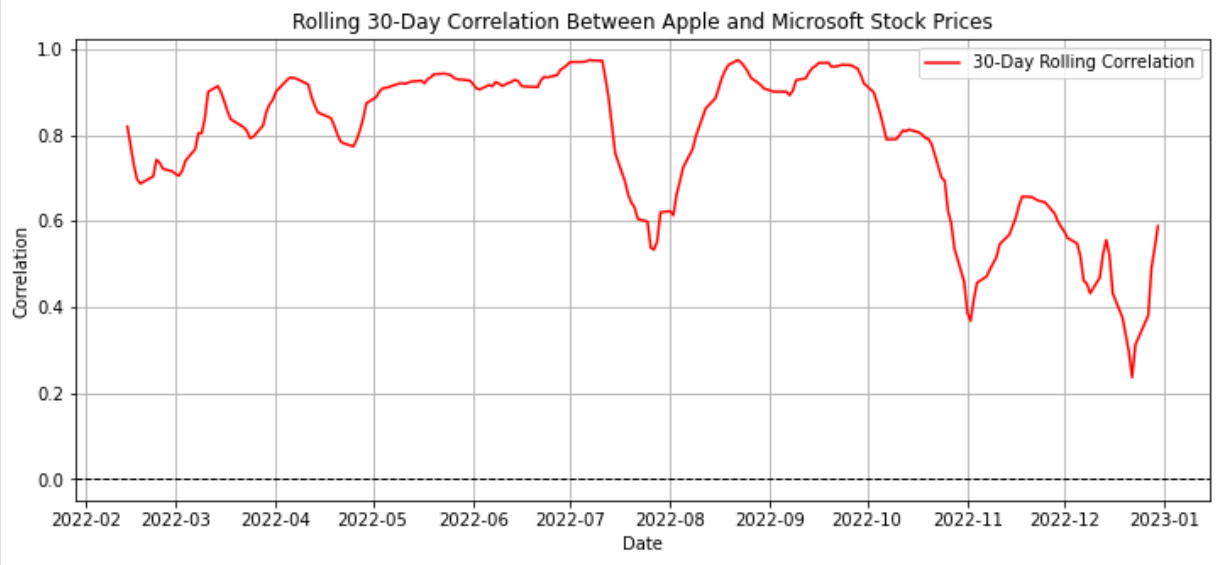
$$\rho(A,B) = \frac{0}{\sqrt{2} * \sqrt{8}} = 0$$

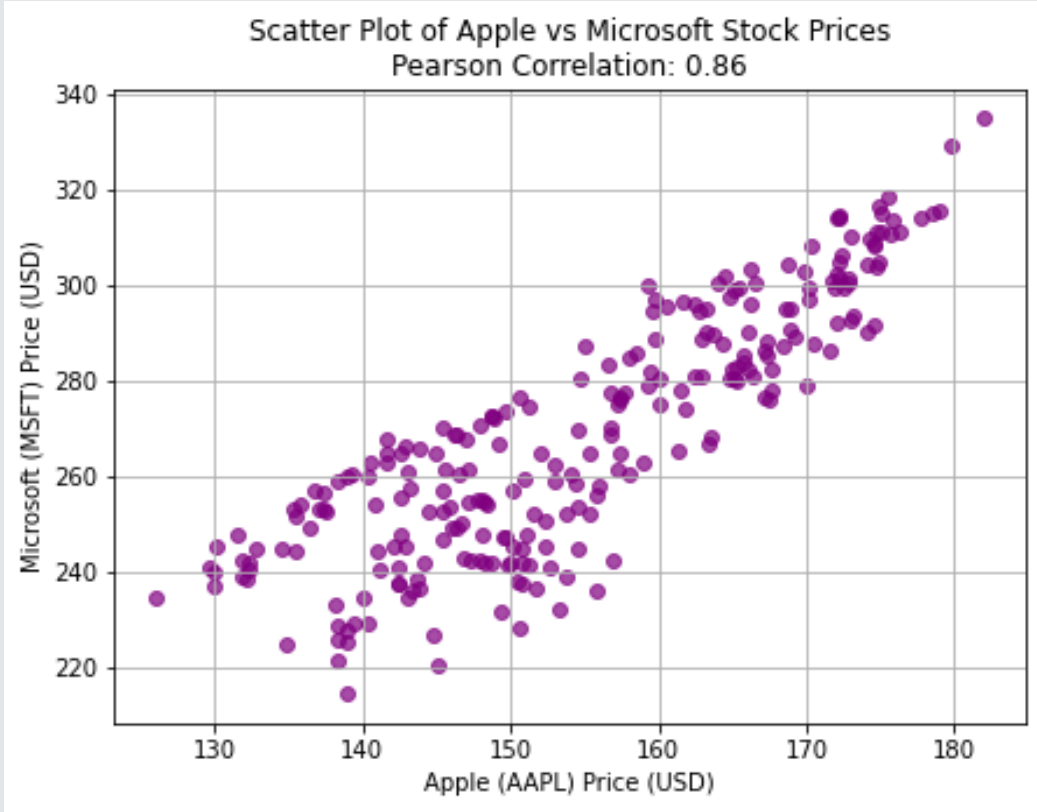
Pearson Correlation – Non linear



Pearson Correlation – Real dataset







Euclidean Distance

- Euclidean distance is the straight-line distance between two points in Euclidean space. It's a popular dissimilarity measure for clustering and nearest-neighbor algorithms.

$$d(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

- **Applications**:
  - *K-means clustering.*
  - *Nearest-neighbor algorithms (e.g., k-NN).*

Euclidean Distance

$$d(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Numerical Exercise: Given $A = [1,2,3]$and $B = [4,5,6]$

Squared Differences:

$$(A_1 - B_1)^2 = (1-4)2 = 9,$$
$$(A_2 - B_2)^2 = (2-5)^2 = 9,$$
$$(A_3 - B_3)^2 = (3-6)^2 = 9$$

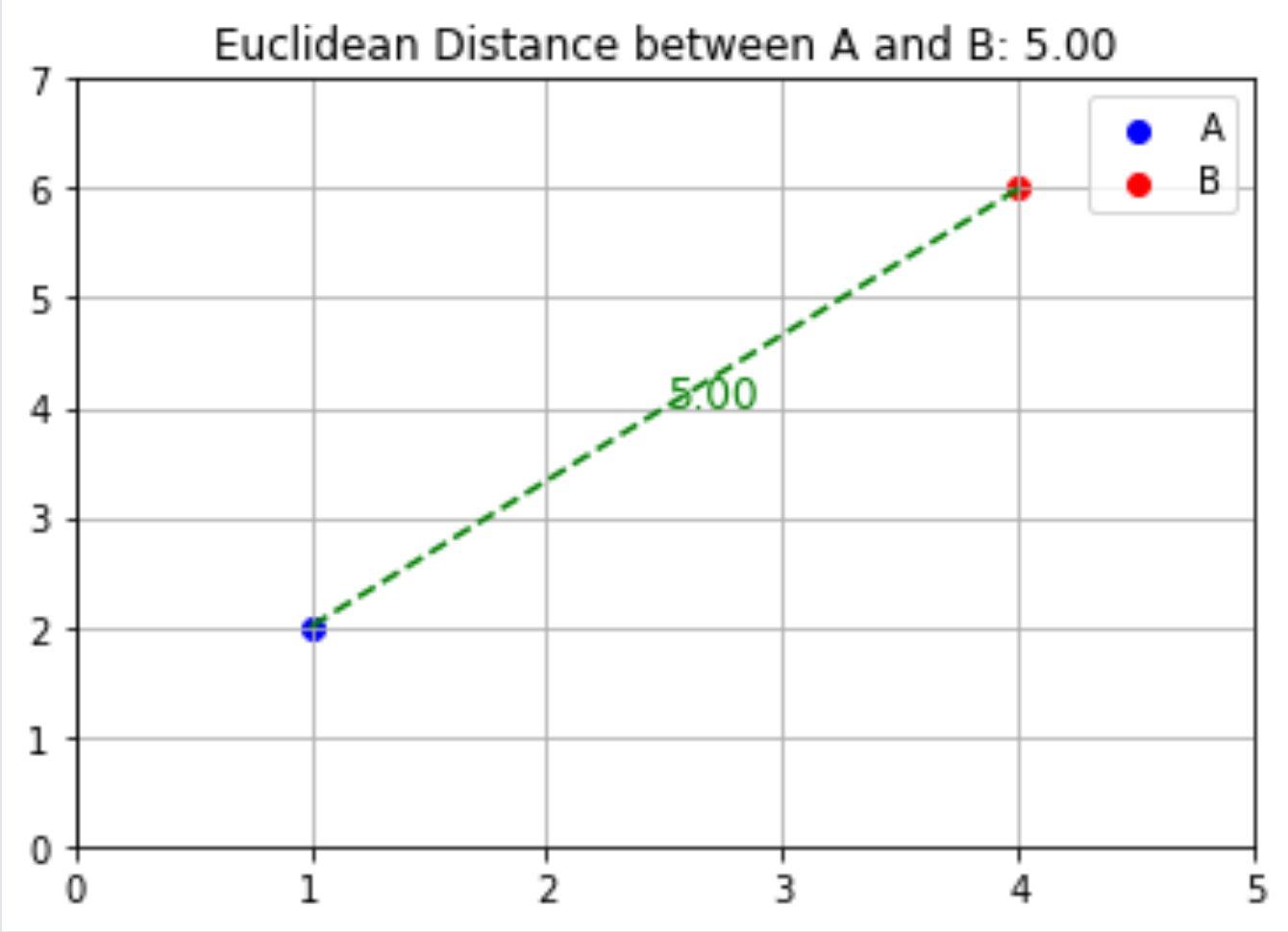
Sum:

$$9 + 9 + 9 = 27$$

Euclidean Distance:

$$d(A,B) = \sqrt{27} \approx 5.196$$

Euclidean Distance



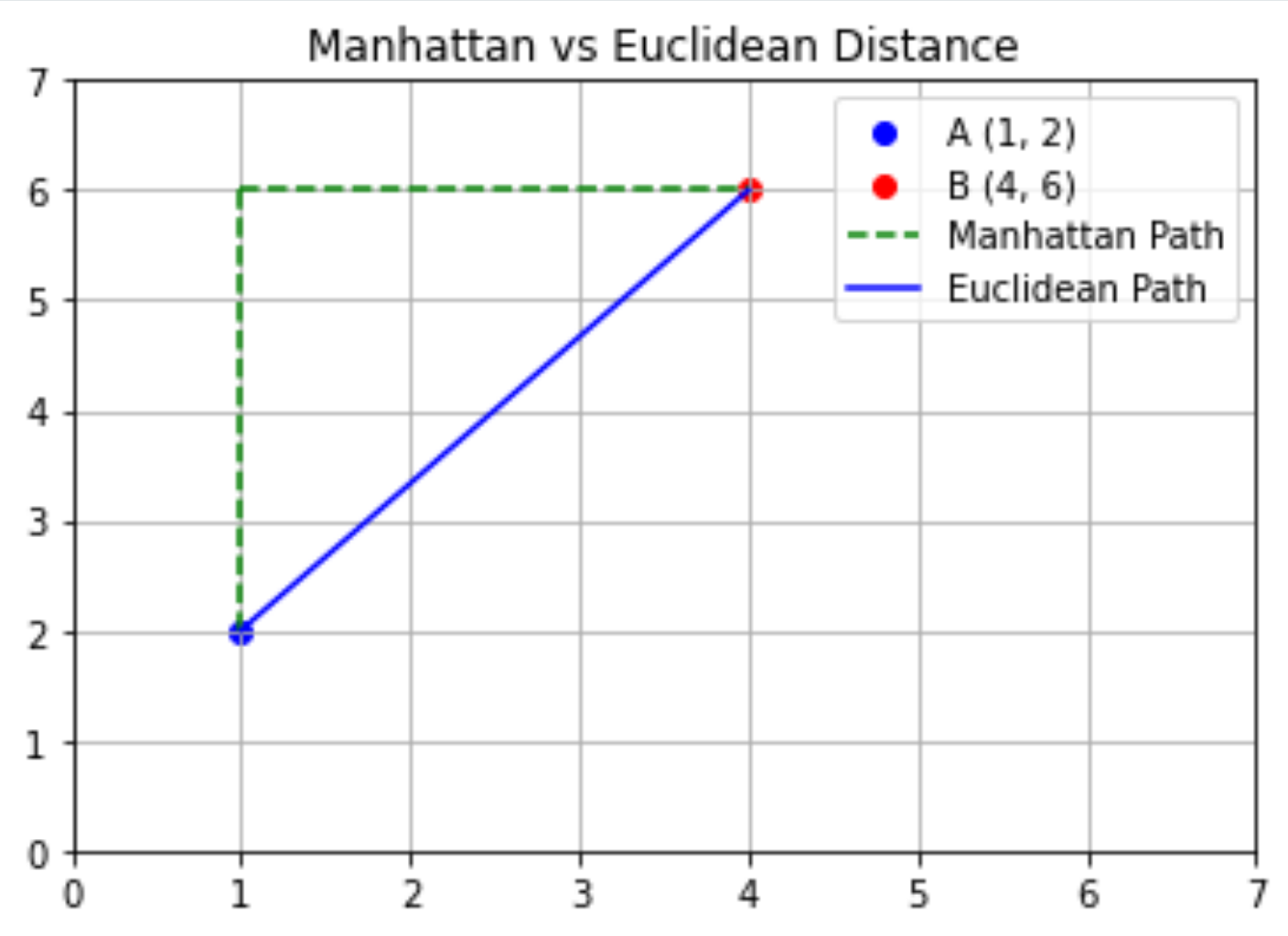
Manhattan Distance

- Manhattan distance measures the distance between two points along axes at right angles. It is defined as the sum of the absolute differences between the coordinates of two points.

$$d(A, B) = \sum_{i=1}^{n} |A_i - B_i|$$

- Euclidean Distance (L2 norm) measures the straight-line (or "as-the-crow-flies") distance between two points in space, while Manhattan Distance measures the total distance one would travel along gridlines. In Manhattan distance, movement is restricted to the grid (like traveling on streets in a city laid out in a grid). Hence, **Manhattan Distance** is more appropriate in scenarios where diagonal movement is not possible or incurs additional cost.

Manhattan Distance



Manhattan Distance Over Euclidean Distance

- Grid-Like Structures:

  Manhattan distance is useful in environments with grid layouts, such as cities with grid-like streets (hence the name "taxicab distance").

  Examples include calculating walking or driving distance in cities where roads are at right angles.

- High-Dimensional Data:

  In high-dimensional spaces, Euclidean distance tends to be less informative because distances between points tend to converge as the number of dimensions increases (this is known as the curse of dimensionality).

  In contrast, Manhattan distance can be more stable in high-dimensional spaces and is often preferred for high-dimensional problems like machine learning (e.g., nearest neighbors, clustering) where axes are treated independently.

Manhattan Distance Over Euclidean Distance

- Feature Independence:

  Manhattan distance is often preferred when features or dimensions are independent of each other. Euclidean distance tends to be more appropriate when dimensions interact or have relationships between them.

- Robustness to Outliers:

  Manhattan distance is less sensitive to outliers compared to Euclidean distance, which can be disproportionately affected by large differences in one dimension.

Applications of Manhattan Distance

- Pathfinding in Grids
- Image Processing
- Clustering Algorithms
- Financial Models

Manhattan Distance

$$d(A,B) = \sum_{i=1}^{n} |A_i - B_i|$$

Numerical Exercise: Given $A = [1,2,3]$and $B = [4,5,6]$

Squared Differences:

$$|A_1 - B_1| = 3,$$
$$|A_2 - B_2| = 3$$
$$|A_3 - B_3| = 3$$

Sum:

$$3 + 3 + 3 = 9$$

Manhattan Distance:

$$d(A,B) = 9$$

Cosine Similarity

- Cosine similarity measures the cosine of the angle between two vectors. This measure focuses on the **direction** rather than the **magnitude** of the vectors. It is widely used in text mining, recommendation systems, and other high-dimensional data spaces.

$$Cosine\ Similarity = \frac{A.B}{||A||||B||}$$

- **Applications**:
  - *Document similarity in information retrieval*
  - *User similarity in recommendation systems*

Cosine Similarity

$$Cosine\ Similarity = \frac{A.B}{||A||||B||}$$

Numerical Exercise: Given $A = [1,2,3]$and $B = [4,5,6]$

Dot Product

$$A.B = 1 * 4 + 2 * 5 + 3 * 6 = 32$$

Magnitude of A

$$||A|| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$

Magnitude of B

$$||B|| = \sqrt{4^2 + 5^2 + 6^2} = \sqrt{77}$$

Cosine Similarity

$$Cosine\ Similarity = \frac{32}{\sqrt{14} * \sqrt{77}} \approx 0.974$$

Cosine Similarity

$$Cosine\ Similarity = \frac{A.B}{||A||||B||}$$

Numerical Exercise: Given $A = [1,2,3]$and $C = [2,4,6]$

Dot Product

$$A.B = 1 * 2 + 2 * 4 + 3 * 6 = 28$$

Magnitude of A

$$||A|| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$

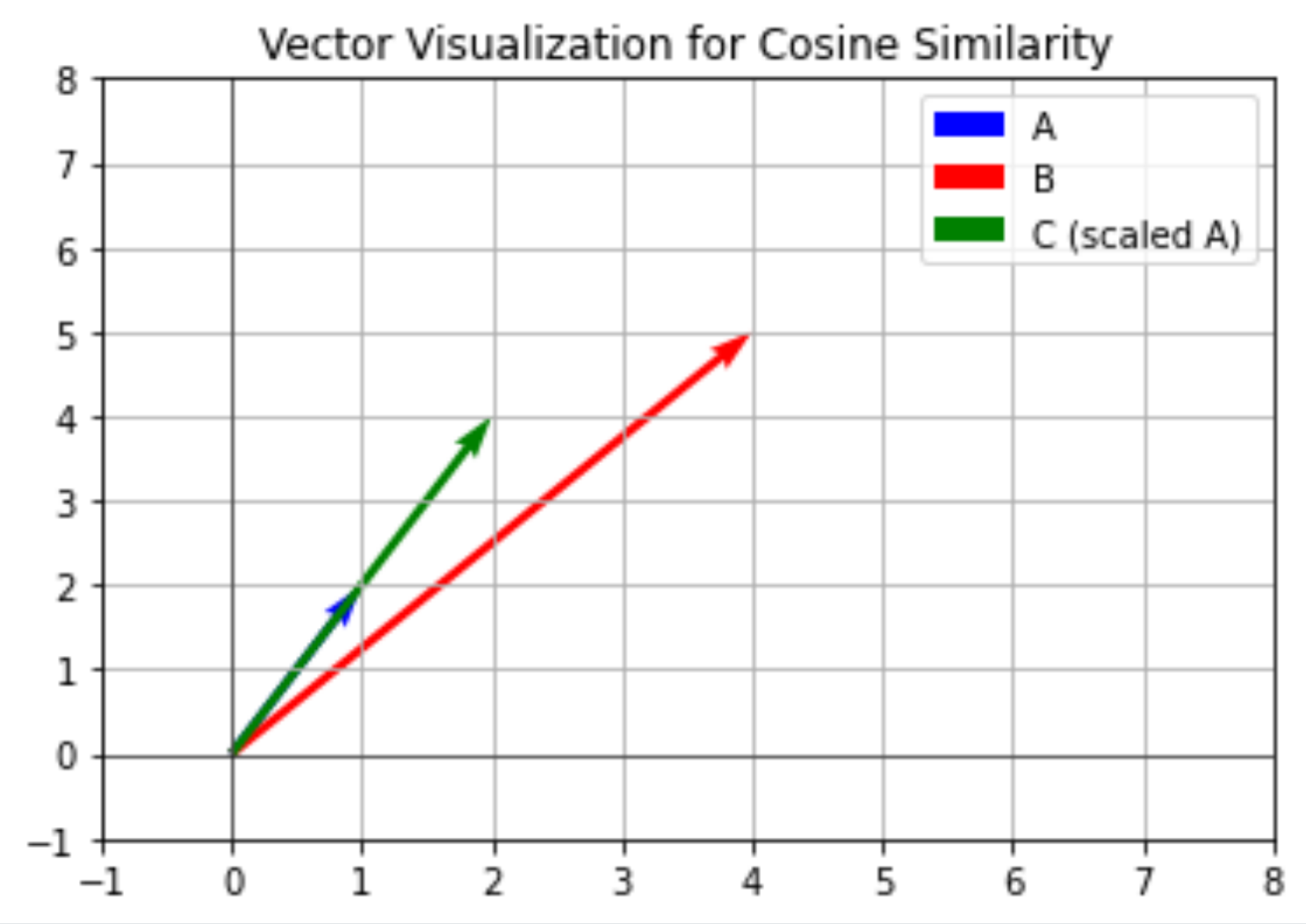
Magnitude of B

$$||B|| = \sqrt{2^2 + 4^2 + 6^2} = \sqrt{56}$$

Cosine Similarity

$$Cosine\ Similarity = \frac{28}{\sqrt{14} * \sqrt{56}} = 1$$

Cosine Similarity



Jaccard Similarity

- Jaccard similarity measures the overlap between two sets. It is commonly used to compare **binary or categorical data** and is particularly useful in text mining and clustering.

$$Jaccard\ Similarity = \frac{|A \cap B|}{|A \cup B|}$$

- Suppose you have two sets:
  - $A = \{1,2,3,4,5\}$ | $B = \{4,5,6,7\}$
- **Intersection** $(A \cap B)$:
  1. *Elements common to both sets: {4,5}. | Size of the intersection: 2.*
- **Union** $(A \cup B)$
  1. *All unique elements from both sets: {1,2,3,4,5,6,7}. | Size of the union: 7.*
- **Jaccard Similarity:**

$$Jaccard\ Similarity(A, B) = 2/7 \approx 0.285$$

Jaccard Similarity - Applications

- **Document Similarity (Text Mining)**
- Given two documents, determine how similar they are based on the words they contain.
  - *Document 1: "Machine learning is amazing." | Document 2: "Deep learning is amazing."*
- Tokenizing each document into words:
  - *Document 1 set: {machine,learning,is,amazing} | Document 2 set: {deep,learning,is,amazing}*
- **Jaccard Similarity**:

$$\frac{|\{learning, is, amazing\}|}{|\{machine, learning, is, amazing, deep\}|} = \frac{3}{5} = 0.6$$

Thus, the documents have a Jaccard similarity of 0.6, meaning they share 60% of their words.

- **Why Jaccard Similarity for Documents?**
  - *It is robust to differences in document length. If two documents share most of their important words, their Jaccard similarity will be high, even if one document is longer than the other.*
  - *It's easy to calculate and works well when documents are treated as unordered collections of words (i.e., sets).*

Jaccard Similarity - Applications

- **Binary Feature Vectors (Machine Learning)**
- In machine learning, features are often represented as binary vectors (0s and 1s) where each element of the vector represents the presence or absence of a feature.
- **Problem**: Compare the similarity between two binary vectors.
- **Solution Using Jaccard Similarity**:
  - *Consider each binary vector as a set of 1s (representing the features that are present).*
  - *Compute the Jaccard similarity to determine the overlap between the sets of 1s in the two vectors.*
- For example:
  - *Vector 1: [1,0,1,0,1](features 1, 3, and 5 are present) | Vector 2: [1,1,0,0,1](features 1, 2, and 5 are present).*
- Intersection (A∩B) = 2 (features 1 and 5). Union (A∪B) = 4 (features 1, 2, 3, and 5).
- **Jaccard Similarity**:

$$\frac{2}{4} = 0.5$$