Linear Algebra

Linear algebra is foundational for data science and machine learning, as many algorithms rely on linear algebra concepts.

Key Concepts in Linear Algebra

- **Scalars, Vectors, Matrices, and Tensors**
  - **Scalars**: Single numbers. {Zero dimension}.
  - Example: s=3.
  - **Vectors**: An array of numbers arranged in a single row or column{one dimension}.
  - **Example : [1,2,3,4,5,6]**
  - **Matrices**: 2-dimensional arrays of numbers.

$$A = \begin{bmatrix} 4 & 9 & 7 \\ 2 & 6 & 1 \end{bmatrix}$$

The Matrix A is of order 2×3

  - **Tensors**: Is a multidimensional array of numerical values with each dimension represent a different mode of direction.

Matrix

- A ***matrix*** is a rectangular array or arrangement of entries or elements displayed in rows and columns put within a square bracket [ ].
- If a matrix A has m rows and n columns, then it is written as

$$A = \left[a_{ij}\right]_{m\times n} \qquad 1 \le i \le m\,, 1 \le j \le n$$

Order of the Matrix

- If a matrix A has m rows and n columns, then the *order or size* of the matrix A is defined to be $m \times n$ (read as *m by n*).

Example

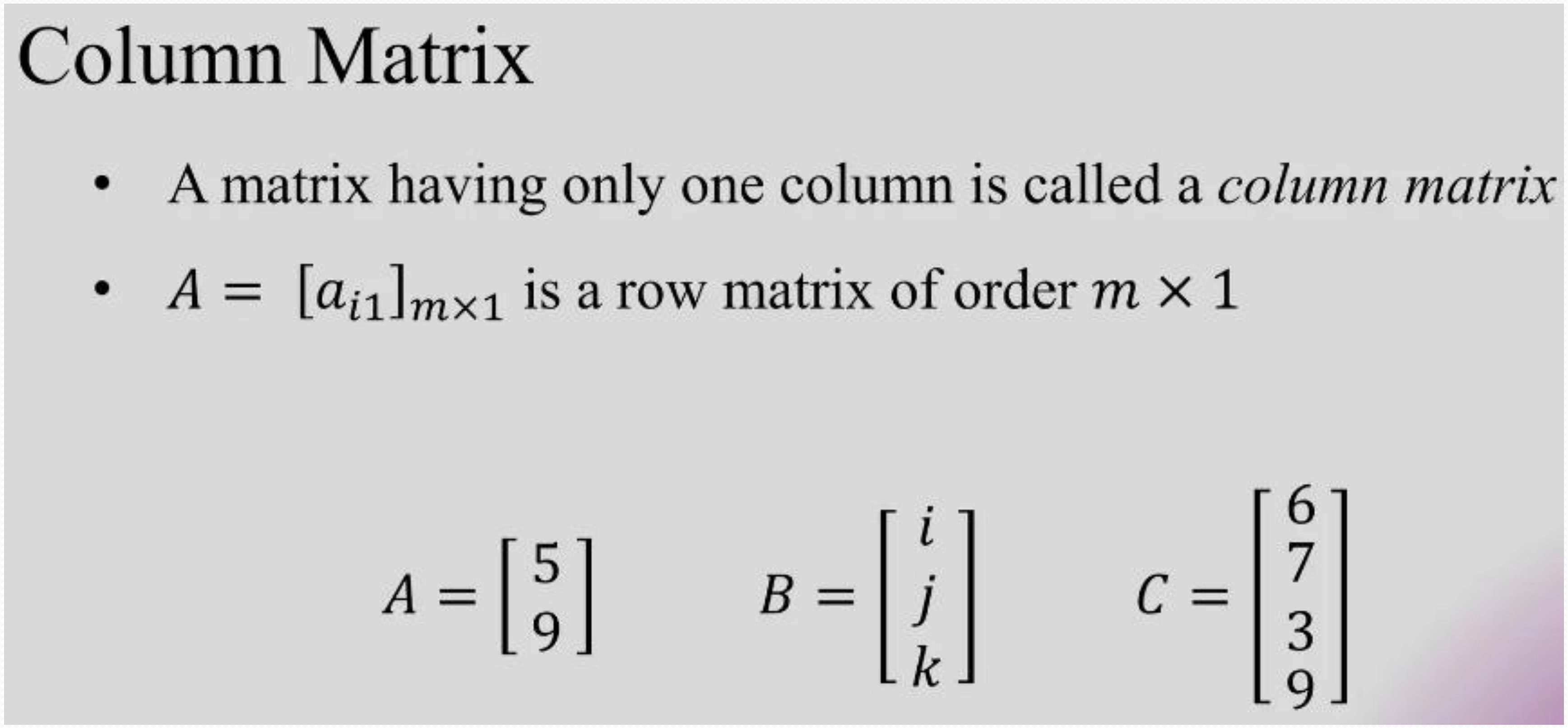
$$A = \begin{bmatrix} 4 & 9 & 7 \\ 2 & 6 & 1 \end{bmatrix}$$

The Matrix A is of order 2×3

Row Matrix

- A matrix having only one row is called a *row matrix*
- $A = \left[a_{1j}\right]_{1\times n}$ is a row matrix of order $1 \times n$

$A = \begin{bmatrix} 4 & 7 \end{bmatrix}$ $B = \begin{bmatrix} 2 & 9 & 3 \end{bmatrix}$ $C = \begin{bmatrix} a & b & c & d \end{bmatrix}$



Column Matrix

- A matrix having only one column is called a *column matrix*
- $A = \ [a_{i1}]_{m\times 1}$ is a row matrix of order $m \times 1$

$$A = \begin{bmatrix} 5 \\ 9 \end{bmatrix} \qquad B = \begin{bmatrix} i \\ j \\ k \end{bmatrix} \qquad C = \begin{bmatrix} 6 \\ 7 \\ 3 \\ 9 \end{bmatrix}$$

Operations on Matrices

- **Matrix Multiplication**:
  - Combining two matrices to produce a new matrix.

Example: $\mathbf{AB}$ where $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}$,

resulting in

$$\mathbf{AB} = \begin{bmatrix} 1 \cdot 2 + 2 \cdot 1 & 1 \cdot 0 + 2 \cdot 2 \\ 3 \cdot 2 + 4 \cdot 1 & 3 \cdot 0 + 4 \cdot 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 10 & 8 \end{bmatrix}$$

Dot Product

- Multiplying corresponding elements of two vectors and summing the results.
- Example: $u \cdot v = u_1 v_1 + u_2 v_2 + \ldots + u_n v_n$

Transpose

- Flipping a matrix over its diagonal.

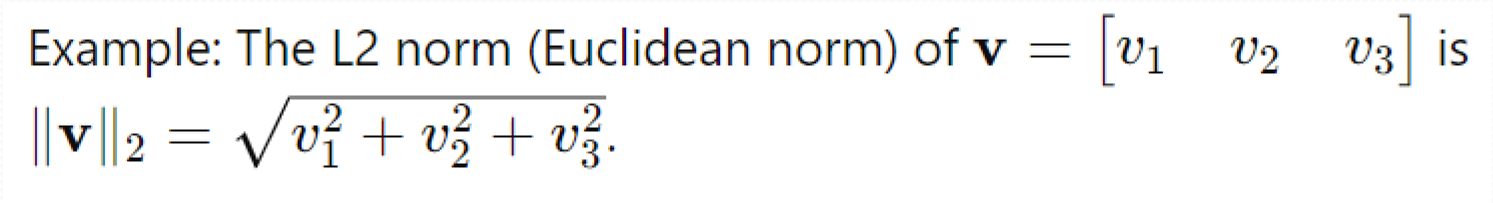
Example: $\mathbf{A}^T$ for $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ results in $\mathbf{A}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$

Eigenvalues and Eigenvectors

- **Eigen vectors**:
  - Vectors that only scale when a matrix is applied to them.
  - Example: For $\mathbf{Av}=\lambda\mathbf{v}$, $\mathbf{v}$ is an eigenvector and λ is the corresponding eigenvalue.
- **Eigen values**:
  - Scalars associated with eigenvectors that indicate the factor by which the eigenvector is scaled.

Norms

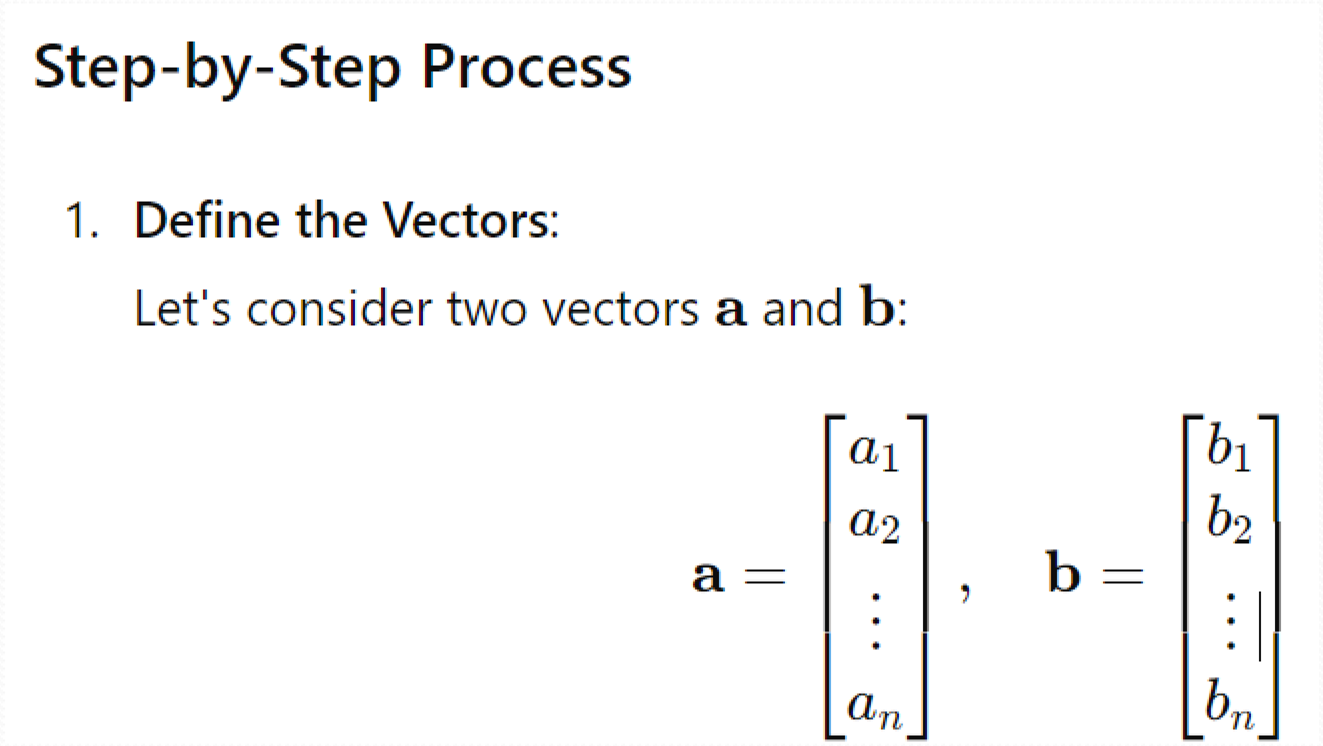
- Measures of the size or length of a vector.



Example: The L2 norm (Euclidean norm) of $\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$ is $\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + v_3^2}$.

Angle between two vectors

- The angle between two vectors can be found using the dot product formula. Specifically, the dot product of two vectors is related to the cosine of the angle between them.



Step-by-Step Process

1. **Define the Vectors:**

   Let's consider two vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

2. **Calculate the Dot Product:**

   The dot product (or inner product) of $\mathbf{a}$ and $\mathbf{b}$ is given by:

$$\mathbf{a} \cdot \mathbf{b} = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

3. **Calculate the Magnitudes:**

   The magnitude (or norm) of a vector $\mathbf{a}$ is given by:

$$||\mathbf{a}|| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$$

   Similarly, the magnitude of $\mathbf{b}$ is:

$$||\mathbf{b}|| = \sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}$$

4. **Calculate the Cosine of the Angle:**

   The cosine of the angle θ between the two vectors is given by:

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

5. **Calculate the Angle:**

   Finally, the angle θ can be found using the inverse cosine (arccos) function:

$$\theta = \cos^{-1}\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}\right)$$

Example

Let's go through an example with two 3-dimensional vectors:

$$\mathbf{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

1. Calculate the Dot Product:

$$\mathbf{a} \cdot \mathbf{b} = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 4 + 10 + 18 = 32$$

2. Calculate the Magnitudes:

$$\|\mathbf{a}\| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{1 + 4 + 9} = \sqrt{14}$$

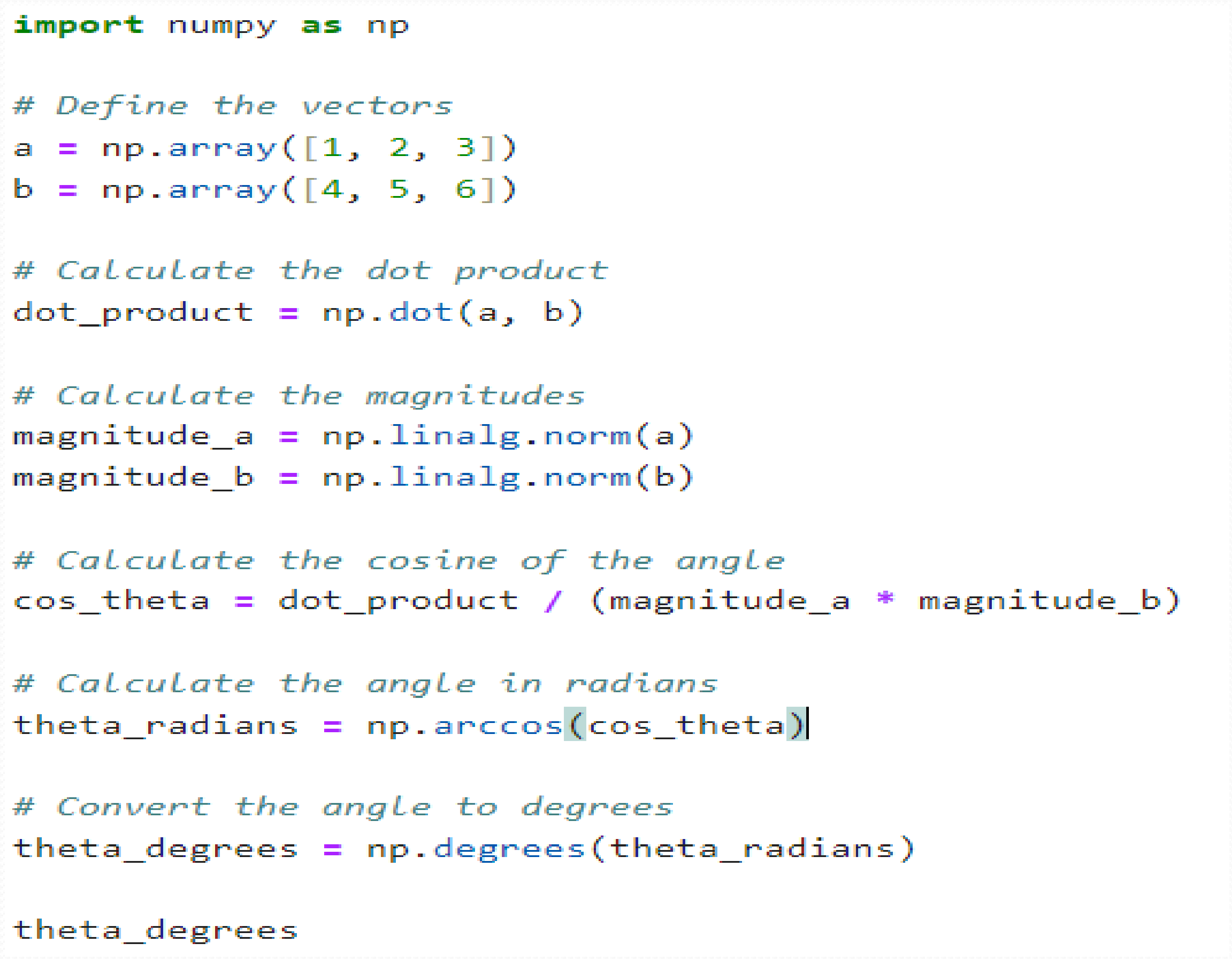
$$\|\mathbf{b}\| = \sqrt{4^2 + 5^2 + 6^2} = \sqrt{16 + 25 + 36} = \sqrt{77}$$

3. Calculate the Cosine of the Angle:

$$\cos(\theta) = \frac{32}{\sqrt{14} \cdot \sqrt{77}} = \frac{32}{\sqrt{1078}}$$

4. Calculate the Angle:

$$\theta = \cos^{-1}\left(\frac{32}{\sqrt{1078}}\right)$$



```python
import numpy as np

# Define the vectors
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# Calculate the dot product
dot_product = np.dot(a, b)

# Calculate the magnitudes
magnitude_a = np.linalg.norm(a)
magnitude_b = np.linalg.norm(b)

# Calculate the cosine of the angle
cos_theta = dot_product / (magnitude_a * magnitude_b)

# Calculate the angle in radians
theta_radians = np.arccos(cos_theta)

# Convert the angle to degrees
theta_degrees = np.degrees(theta_radians)

theta_degrees
```

Cosine Similarity

- Cosine similarity is a measure that calculates the cosine of the angle between two non-zero vectors in a multidimensional space.
- This metric is often used to measure the similarity between documents or data points in high-dimensional spaces such as in text analysis, clustering, and other machine learning applications.
- It is used to determine how similar two vectors are regardless of their magnitude.

Cosine Similarity Formula

- The cosine similarity between two vectors **A** and **B** is given by:

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A||\,||B||}$$

- Where A·B is the dot product of vectors A and B, and ||A|| and ||B|| are the magnitudes (norms) of the vectors.

Applications of Cosine Similarity

- **Text Box:**
  - **Text Analysis:** Measuring similarity between documents or text segments based on their term vectors.
- **Clustering:**
  - Grouping similar data points together based on their vector representation.
- **Recommendation Systems:**
  - Comparing user preferences or item features to suggest similar items.

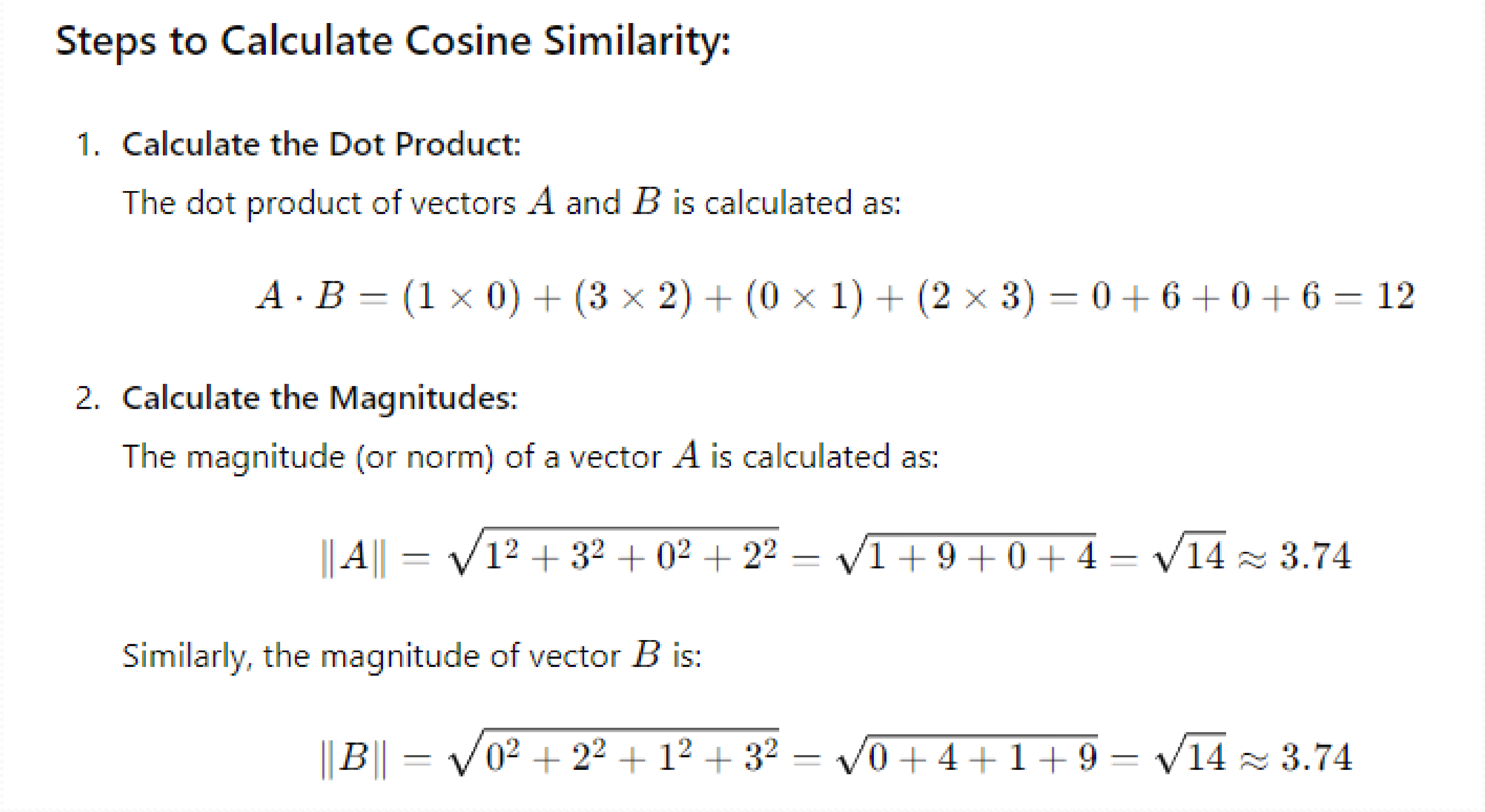
Example of Cosine Similarity

- Consider two text documents represented as term frequency vectors. These vectors represent the frequency of specific terms (words) in each document. Let's calculate the cosine similarity between these two vectors to determine how similar the documents are.
- **Document Vectors:**

  Let's assume the documents are vectorized as follows:

  Document A: [1,3,0,2]

  Document B: [0,2,1,3]
- These vectors could represent the term frequencies of four specific words in two different documents.



Steps to Calculate Cosine Similarity:

1. **Calculate the Dot Product:**

   The dot product of vectors A and B is calculated as:

$$A \cdot B = (1 \times 0) + (3 \times 2) + (0 \times 1) + (2 \times 3) = 0 + 6 + 0 + 6 = 12$$

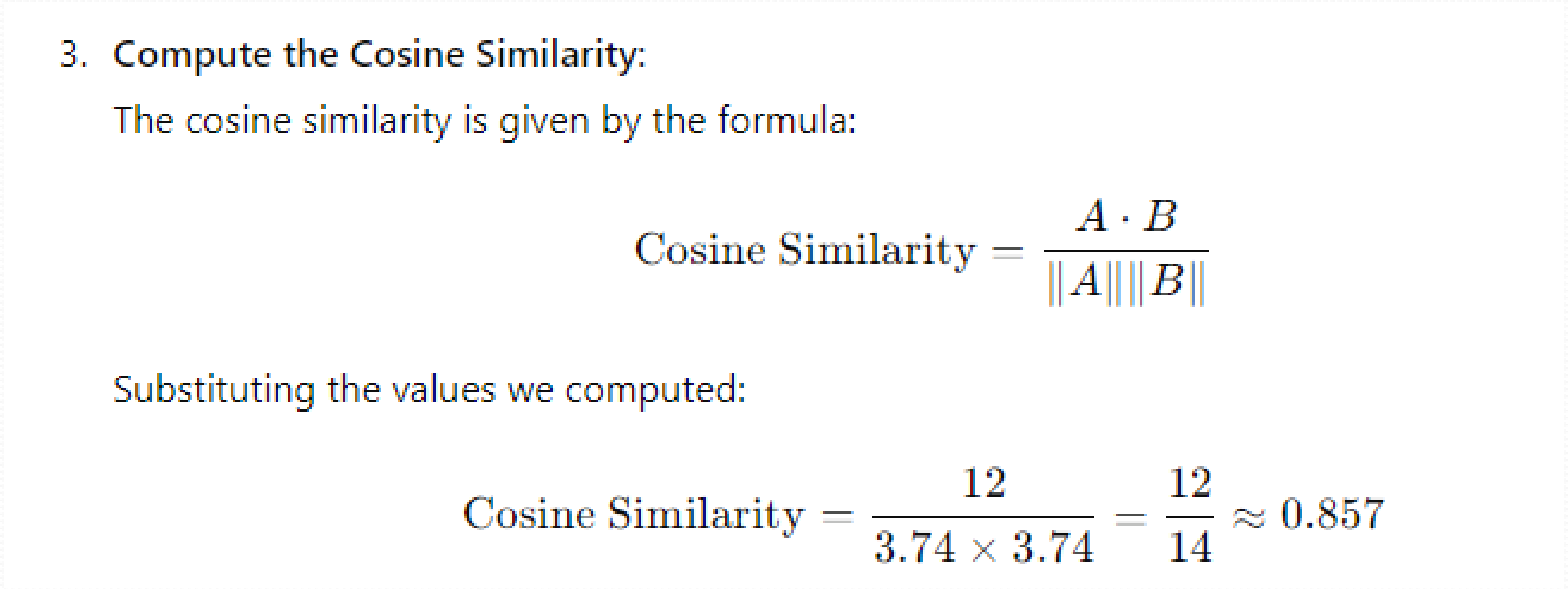
2. **Calculate the Magnitudes:**

   The magnitude (or norm) of a vector A is calculated as:

$$\|A\| = \sqrt{1^2 + 3^2 + 0^2 + 2^2} = \sqrt{1 + 9 + 0 + 4} = \sqrt{14} \approx 3.74$$

   Similarly, the magnitude of vector B is:

$$\|B\| = \sqrt{0^2 + 2^2 + 1^2 + 3^2} = \sqrt{0 + 4 + 1 + 9} = \sqrt{14} \approx 3.74$$



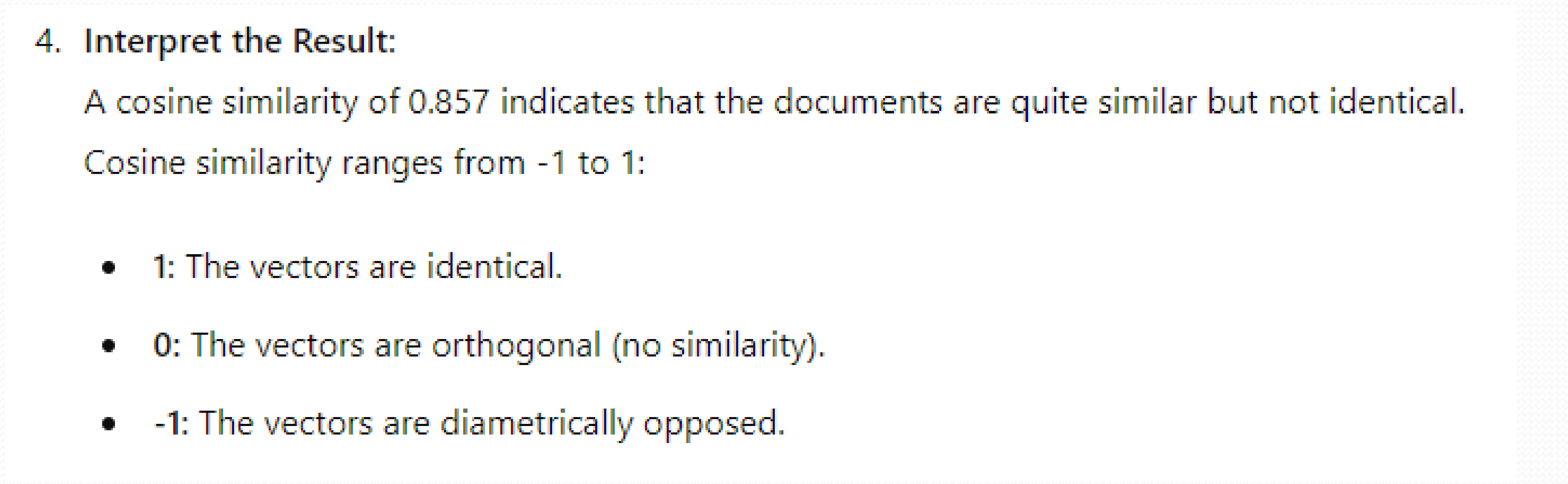
3. **Compute the Cosine Similarity:**

   The cosine similarity is given by the formula:

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A||\,||B||}$$

   Substituting the values we computed:

$$\text{Cosine Similarity} = \frac{12}{3.74 \times 3.74} = \frac{12}{14} \approx 0.857$$



4. **Interpret the Result:**

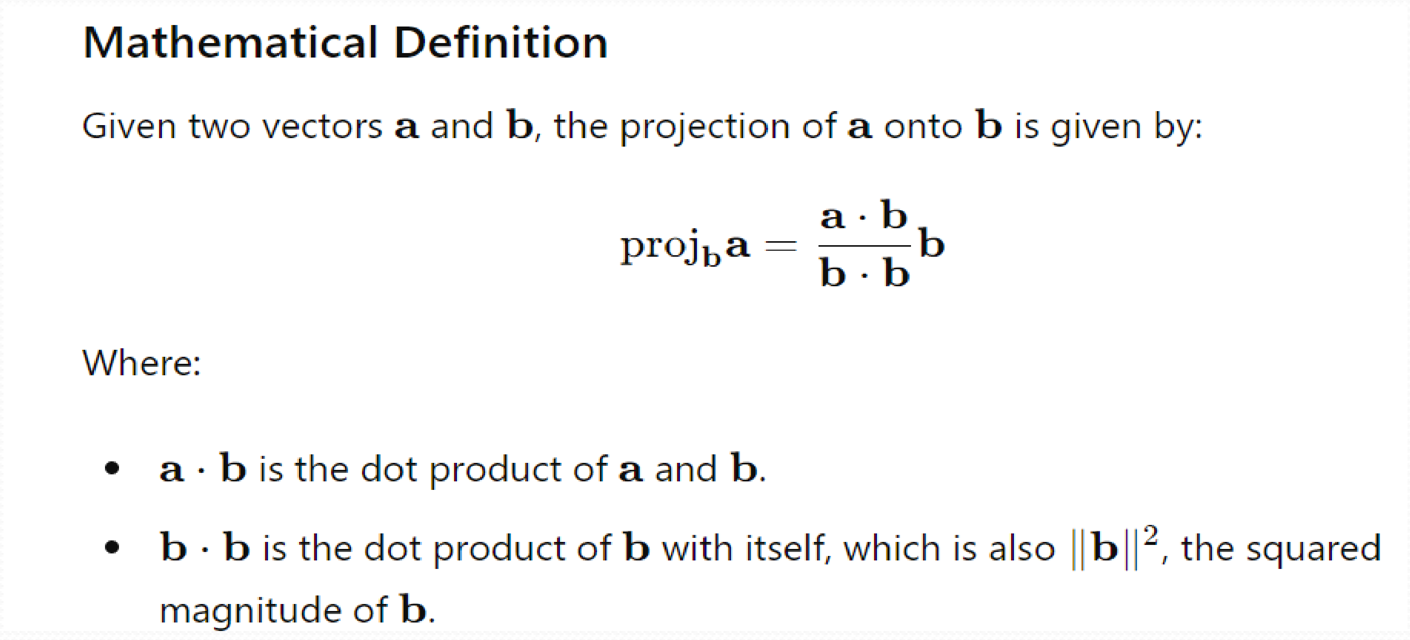
   A cosine similarity of 0.857 indicates that the documents are quite similar but not identical.

   Cosine similarity ranges from -1 to 1:

   - **1**: The vectors are identical.
   - **0**: The vectors are orthogonal (no similarity).
   - **-1**: The vectors are diametrically opposed.

Projection of a Vector

- The projection of one vector onto another is a fundamental concept in linear algebra that has significant applications in data science and machine learning.
- The projection of vector **a** onto vector **b** is a vector that represents the shadow or footprint of **a** in the direction of **b**.



Mathematical Definition

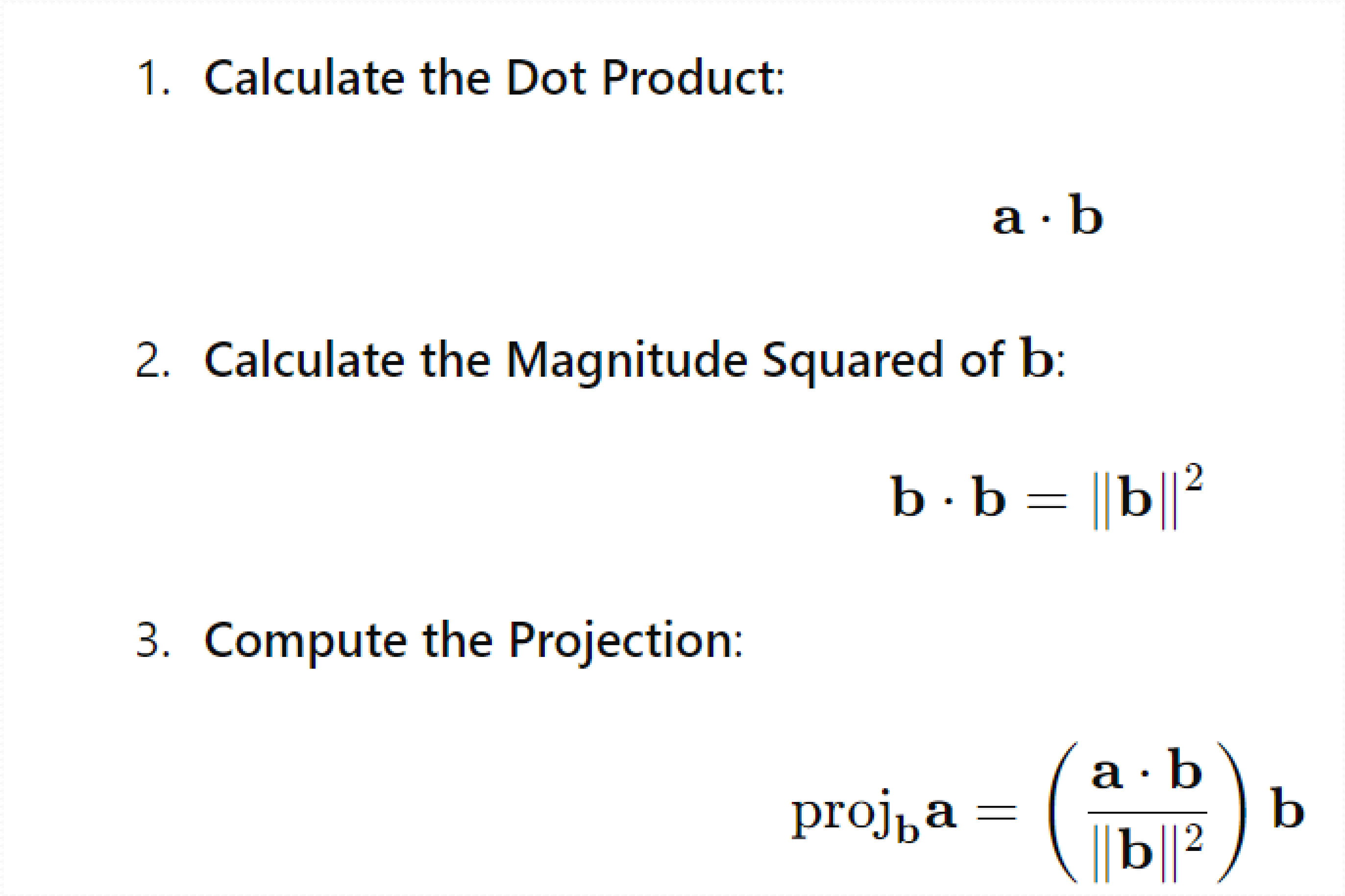
Given two vectors $\mathbf{a}$ and $\mathbf{b}$, the projection of $\mathbf{a}$ onto $\mathbf{b}$ is given by:

$$\text{proj}_{\mathbf{b}}\mathbf{a} = \frac{\mathbf{a} \cdot \mathbf{b}}{\mathbf{b} \cdot \mathbf{b}}\mathbf{b}$$

Where:

- $\mathbf{a} \cdot \mathbf{b}$ is the dot product of $\mathbf{a}$ and $\mathbf{b}$.
- $\mathbf{b} \cdot \mathbf{b}$ is the dot product of $\mathbf{b}$ with itself, which is also $||\mathbf{b}||^2$, the squared magnitude of $\mathbf{b}$.

Steps to Compute the Projection



1. Calculate the Dot Product:

$$\mathbf{a} \cdot \mathbf{b}$$

2. Calculate the Magnitude Squared of $\mathbf{b}$:

$$\mathbf{b} \cdot \mathbf{b} = \|\mathbf{b}\|^2$$

3. Compute the Projection:

$$\text{proj}_{\mathbf{b}}\mathbf{a} = \left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|^2}\right)\mathbf{b}$$

Example

Let's consider vectors $\mathbf{a} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

1. Calculate the Dot Product:

$$\mathbf{a} \cdot \mathbf{b} = 3 \cdot 1 + 4 \cdot 2 = 3 + 8 = 11$$

2. Calculate the Magnitude Squared of $\mathbf{b}$:

$$\mathbf{b} \cdot \mathbf{b} = 1^2 + 2^2 = 1 + 4 = 5$$

3. Compute the Projection:

$$\text{proj}_{\mathbf{b}}\mathbf{a} = \left(\frac{11}{5}\right) \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{11}{5} \\ \frac{22}{5} \end{bmatrix} = \begin{bmatrix} 2.2 \\ 4.4 \end{bmatrix}$$

Practical Application in Machine Learning/Data Science

- **Data Representation**:
  - **Linear Transformations**: Dataset and features in machine learning models is often represented as vectors and matrices.
  - Each row of a matrix could represent a data sample, and each column a feature.
  - In natural language processing (NLP), text data is converted into vectors (word embeddings) that capture semantic meanings.
  - Matrix operations are then used to analyze and transform these vectors.
  - Example: Scaling and rotating data points using matrix multiplication.

- **Model Training**:
  - Use matrix operations for efficient computation of gradients and updates.
  - Example: Gradient descent algorithm involves matrix operations to update weights.
- **Dimensionality Reduction**:
  - Techniques like PCA (Principal Component Analysis) use eigenvalues and eigenvectors to reduce dimensions.
  - Example: PCA transforms the data into a lower-dimensional space while preserving variance.

- **Regularization**:
  - Apply norms to weights in regularization techniques to prevent overfitting.
  - Example: L2 regularization adds a penalty equal to the square of the magnitude of coefficients.
- **Matrices in Image Processing:**
  - Matrices represent images in data science.
  - Each pixel of an image can be considered an element in a matrix.
  - Operations like blurring, edge detection, and color transformation are performed using matrix manipulations.

Special Types of Matrices

- **Diagonal Matrix**: Non-zero entries only on the main diagonal.
- **Orthogonal Matrix**: Transpose equals its inverse.
- **Symmetric Matrix**: Equal to its transpose.
- **Sparse Matrix**: Most elements are zero, important for efficiency in large datasets.

Special Types of Matrices

- **Diagonal Matrix:**
  - **Definition:** A diagonal matrix has non-zero entries only on its main diagonal, with all off-diagonal elements being zero.
  - **In Data Science:**
    - **Principal Component Analysis (PCA):** Eigenvalues of covariance matrices, often used in PCA for feature reduction, are represented as diagonal matrices. This simplifies the process of projecting data into principal components.
    - **Scaling Factors:** Diagonal matrices can represent scaling factors in different dimensions, aiding in normalization or scaling operations.

- Example: $\begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \end{pmatrix}$

- **Orthogonal Matrix:**
  - **Definition:** An orthogonal matrix has the property that its transpose is equal to its inverse. This means multiplying an orthogonal matrix by its transpose yields the identity matrix.
  - **In Data Science:**
  - **Data Transformation:** Orthogonal matrices are used in transforming data while preserving distances and angles, which is essential in algorithms like PCA, where data is rotated into a new coordinate system without distorting relationships.
  - **Rotation and Reflection:** These matrices are used for rotation and reflection transformations in space, preserving the magnitude of vectors, which is important in image processing and computer vision tasks.

$$Q = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Symmetric Matrix:

- **Definition:** A symmetric matrix is equal to its transpose, meaning the element in row i and column j is the same as the element in row j and column i.

$$S = \begin{pmatrix} 1 & 7 & 3 \\ 7 & 4 & -5 \\ 3 & -5 & 6 \end{pmatrix}$$

- **In Data Science:**
  - **Covariance and Correlation Matrices:** These matrices, which measure relationships between variables, are symmetric.
  - Symmetric matrices are crucial in understanding data distributions and dependencies in statistical modeling.

$$\mathrm{Cov}(X, Y) = \begin{pmatrix} \sigma_{XX} & \sigma_{XY} \\ \sigma_{YX} & \sigma_{YY} \end{pmatrix}$$

- **Sparse Matrix:**
  - **Definition:** A sparse matrix is one in which most of the elements are zero. This contrasts with dense matrices, which have few zero elements.

$$\text{Sparse Matrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 8 & 0 \end{pmatrix}$$

  - **In Data Science:**
    - **Memory and Computational Efficiency:** Sparse matrices save memory and computational resources by storing only non-zero elements. This is crucial for handling large-scale data, such as in natural language processing (NLP) and graph algorithms.
    - **Recommendation Systems:** In collaborative filtering and recommendation systems, the user-item interaction matrix is often sparse. Efficiently handling and computing with sparse matrices is essential for scalable recommendations.
    - **Image Processing:** Sparse matrices represent images or signals with lots of zero-valued pixels or components, making them efficient for storing and processing in compressed formats.