Linear Algebra and Calculus for ML

- **Essential Linear Algebra Concepts**
  - It's the backbone of machine learning
    - Extremely important for anyone who wants enter the world of ML
    - It provides mathematical frameworks for many algorithms and techniques
    - Also - helps manipulate and handle data, stored in the form of Vectors & Matrices
    - Therefore, it is useful in the data preprocessing as well as the model building
  - Scalars
    - Single numerical values, without direction, magnitude only $k = 3$
    - Used to scale vectors or matrices through operations like multiplication
  - Vectors
    - Quantities that have both magnitude and direction $\mathbf{v} = \begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix}$
    - Represented as arrows in space
  - Matrices
    - Rectangular arrays of numbers, arranged in rows and columns. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
    - Used to represent linear transformations, systems of linear equations, and data transformations in machine learning.
  - Operations
    - Addition and Subtraction
      - Adding or subtracting corresponding elements
        - addition: $[\mathbf{u} + \mathbf{v} = \begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 2+3 \\ -1+0 \\ 4+(-2) \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ 2 \end{bmatrix}]$
        - subtraction: $[\mathbf{u} - \mathbf{v} = \begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix} - \begin{bmatrix} 3 \\ 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 2-3 \\ -1-0 \\ 4-(-2) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 6 \end{bmatrix}]$
    - Scalar multiplication
      - Multiplying each element of a vector or matrix by a scalar $k \cdot \mathbf{v} = 3 \cdot \begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \cdot 2 \\ 3 \cdot (-1) \\ 3 \cdot 4 \end{bmatrix} = \begin{bmatrix} 6 \\ -3 \\ 12 \end{bmatrix}$
    - Dot product
      - Measures the similarity of their directions
      - Multiplying corresponding elements of two vectors and summing the results $(\mathbf{u} = [u_1, u_2, u_3]$ and $\mathbf{v} = [v_1, v_2, v_3])$
        - $\mathbf{u} \cdot \mathbf{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + u_3 \cdot v_3$
    - Matrix Multiplication
      - Used in machine learning for various tasks
        - Transformation of feature vectors
        - Computation of model parameters
        - Neural network operations
          - Feedforward
          - Backpropagation
      - We have to perform row-column multiplication. $[A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ and $B = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}]$
        - $[C = \begin{bmatrix} 7 & 2 \\ 5 & 4 \end{bmatrix}]$
  - Let's try using NumPy!
- **Gradient Descent Optimization**
  - Moldels have objective functions
    - Measure model performance on a task. Eg: Minimizing the error on a training dataset
  - Optimiazation
    - Process of adjusting parameters of a model to minimize an objective function
  - Gradient Descent - Powerful optimization technique
    - Goal - To find minimum of functions
      - To find the point where the function has the lowest value
        - eg: for f(x) = x^2, the point is x = 0.
  - Update rule
    - Gradient descent uses the gradient to iteratively move towards the minimum $x_{\text{new}} = x_{\text{old}} - \text{learning\_rate} \times \text{gradient}(x_{\text{old}})$
      - x_new - updated value of x
      - x_old - current value of x
      - learning rate - a small positive number
        - the step size of the algorithm
          - too small - slow algorithm
          - too large - might overshoot the minimum
      - gradient - the derivative of the old function
- **Project: Implementing Gradient Descent to Find the Minimum of a Simple Quadratic Function**
  - Implementation of gradient descent to find the minimum of a quadratic function.
  - $f(x) = (x-3)^2 + 4.$