Algorithm: Principal Component Analysis (PCA)

Input: Dataset X with m samples and n features

Output: Principal components and transformed dataset

Step 1: Standardize the Dataset

- Compute the mean of each feature:

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} X_{ij}, \quad \text{for each feature } j$$

- Subtract the mean from each feature to center the data:

$$X' = X - \mu$$

- If required, normalize each feature by dividing by its standard deviation.

Step 2: Compute the Covariance Matrix

- Compute the covariance matrix C of the standardized dataset:

$$C = \frac{1}{m-1} X'^T X'$$

where C is an $n \times n$ matrix.

Step 3: Compute Eigenvalues and Eigenvectors

- Compute the eigenvalues λ and eigenvectors v of the covariance matrix:

$$Cv = \lambda v$$

- The eigenvectors represent the principal components.

Step 4: Select the Top k Principal Components

- Sort the eigenvalues in descending order.
- Choose the top k eigenvectors corresponding to the largest k eigenvalues.

Step 5: Transform the Dataset

- Form the projection matrix W using the selected eigenvectors.
- Transform the original dataset using:

$$X_{\text{reduced}} = X'W$$

where X_{reduced} is the transformed dataset with k dimensions.

KNN

Algorithm: K-Nearest Neighbors (KNN)

Input:

- Training dataset X_{train} with labels Y_{train}
- Test instance X_{test}
- Number of neighbors k
- Distance metric (e.g., Euclidean, Manhattan)

Output:

- Predicted label for X_{test}

Step 1: Compute Distance

- For each training instance X_i, compute the distance between X_{test} and X_i:
  - **Euclidean Distance** (most common):

$$d(X_{\text{test}}, X_i) = \sqrt{\sum_{j=1}^{n} (X_{\text{test},j} - X_{i,j})^2}$$

  - Other possible distance metrics: Manhattan, Minkowski, Cosine Similarity, etc.

Step 2: Find the k Nearest Neighbors

- Sort the distances in ascending order.
- Select the top k nearest points.

Step 3: Assign a Class (For Classification)

- Take a **majority vote** among the k nearest neighbors.
- Assign the most frequent class label to X_{test}.

For Regression:

- Compute the **average** of the target values of the k nearest neighbors.

Step 4: Return the Predicted Class or Value

- Output the most frequent class (for classification) or the mean value (for regression).

LDA

Algorithm: Linear Discriminant Analysis (LDA)

Input:

- Training dataset X with m samples and n features
- Class labels Y
- Number of classes c

Output:

- Projection matrix W
- Transformed dataset X_{LDA}

Step 1: Compute Class-wise Mean Vectors

For each class k, compute the mean vector:

$$\mu_k = \frac{1}{N_k} \sum_{i \in C_k} X_i$$

where N_k is the number of samples in class k, and C_k represents the set of indices belonging to class k.

Compute the overall mean:

$$\mu = \frac{1}{m} \sum_{i=1}^{m} X_i$$

Step 2: Compute Scatter Matrices

2.1 Compute Within-Class Scatter Matrix S_W

$$S_W = \sum_{k=1}^{c} \sum_{i \in C_k} (X_i - \mu_k)(X_i - \mu_k)^T$$

Alternatively, using covariance for each class:

$$S_W = \sum_{k=1}^{c} S_k$$

where

$$S_k = \sum_{i \in C_k} (X_i - \mu_k)(X_i - \mu_k)^T$$

2.2 Compute Between-Class Scatter Matrix S_B

$$S_B = \sum_{k=1}^{c} N_k (\mu_k - \mu)(\mu_k - \mu)^T$$

Step 3: Solve the Generalized Eigenvalue Problem

Find the eigenvalues λ and eigenvectors v by solving:

$$S_W^{-1} S_B v = \lambda v$$

The eigenvectors form the **discriminant directions**.

Step 4: Select the Top k Discriminant Components

- Sort eigenvectors by their corresponding eigenvalues in descending order.
- Select the top k eigenvectors to form the **projection matrix** W.

Step 5: Transform the Dataset

$$X_{\text{LDA}} = XW$$

where X_{LDA} is the transformed dataset with reduced dimensions.

LR

Algorithm: Linear Regression (Analytical Solution - Least Squares Estimation)

Input:

- Feature values X (independent variable)
- Target values T (dependent variable)

Output:

- Optimal slope W_1 and intercept W_0
- Predicted values $\hat{T}$

Step 1: Compute Means

Calculate the mean of X and T:

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$$

$$\bar{T} = \frac{1}{m}\sum_{i=1}^{m} T_i$$

Step 2: Compute Slope W_1

$$W_1 = \frac{\overline{XT} - \bar{X}\bar{T}}{\overline{X^2} - \bar{X}^2}$$

where

$$\overline{XT} = \frac{1}{m}\sum_{i=1}^{m} X_i T_i, \quad \overline{X^2} = \frac{1}{m}\sum_{i=1}^{m} X_i^2$$

Step 3: Compute Intercept W_0

$$W_0 = \bar{T} - W_1\bar{X}$$

Step 4: Compute Predictions

$$\hat{T} = W_0 + W_1 X$$