# Feature Engineering

1. **Feature Transformation:**
   * Modifying the scale or distribution of features, Handling Missing values, Outlier treatment, Handling categorical values.
2. **Feature Construction:**
   * Creating new features from existing ones. This may involve combining or transforming features to capture additional information.
3. **Feature Selection:**
   * Choosing a subset of relevant features to use in the model. This helps in reducing dimensionality and improving model performance.
4. **Feature Extraction:**
   * Deriving new features from the original ones, often using mathematical techniques. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and t-distributed Stochastic Neighbour Embedding (t-SNE) are examples.

# Curse of Dimensionality:

**Definition**: Refers to the challenges that arise when dealing with datasets with a large number of features, especially when many of those features are irrelevant or redundant.

**Issues**: High-dimensional spaces can lead to increased computational complexity, increased risk of overfitting, and difficulties in visualizing and interpreting the data.

## Dimensionality Reduction:

* **Purpose:**
  + To overcome the curse of dimensionality by reducing the number of features while preserving the most important information.
* **Techniques:**
  + **Feature Selection:**
    - Forward Selection: Adds one feature at a time, starting with the most significant.
    - Backward Elimination: Removes one feature at a time, starting with the least significant.
  + **Feature Extraction:**
    - Principal Component Analysis (PCA): Linear technique that transforms the data into a new coordinate system where the axes are the principal components.
    - Linear Discriminant Analysis (LDA): Supervised technique that maximizes the separation between classes.
    - t-distributed Stochastic Neighbour Embedding (t-SNE): Non-linear technique for visualization and dimensionality reduction.

# PCA

Principal Component Analysis (PCA) is indeed an unsupervised machine learning algorithm used for dimensionality reduction and feature extraction

Objectives:

* **Dimensionality reduction**: PCA aims to reduce the dimensionality of a dataset while retaining as much of its original **variability** as possible.
* **Feature extraction**: It identifies the most important features in the data and represent them as principal components.

## Process of PCA:

* **Standardization**: Standardize the data to have a mean of 0 and a standard deviation of 1. This is important for ensuring that all features contribute equally to analysis.
* **Covariance Matrix**: Calculate the covariance matrix for the standardized data. The covariance matrix describes the relationship between different features.
* **Eigen decompositions**: Perform Eigen decomposition on the covariance matrix to obtain eigen vectors and eigen values. Eigenvectors represent the directions of maximum variance in the data, and eigenvalues indicate the magnitude of variance along those directions.
* **Selection of Principal components**: Sort the eigenvectors based on their corresponding eigenvalues in decreasing order. Select the top k eigenvectors to form the principal components, where k is the desired lower dimensionality
* **Projection**: Project the original data onto the space defined by the selected principal components. This results in a new representation of the data in lower-dimensional space.

## Advantages of PCA:

* Reduction of dimensionality:
  + PCA allows for the reduction of the number of features while retaining the most important information
* Noise reduction:
  + By focusing on the principal components with the highest variance, PCA can help reduce the impact of noise in the data.
* Visualization:
  + Lower dimensional representations obtained through PCA can be easily visualized, aiding in data exploration and interpretation.

## Eigen values and Eigen vectors:

Eigen values are associated with Eigenvectors in Linear algebra. Both the terms are used in the analysis of linear transformations. Eigenvalues are special set of scalar values that is associated with the set of linear equations most probably in the matrix equations.

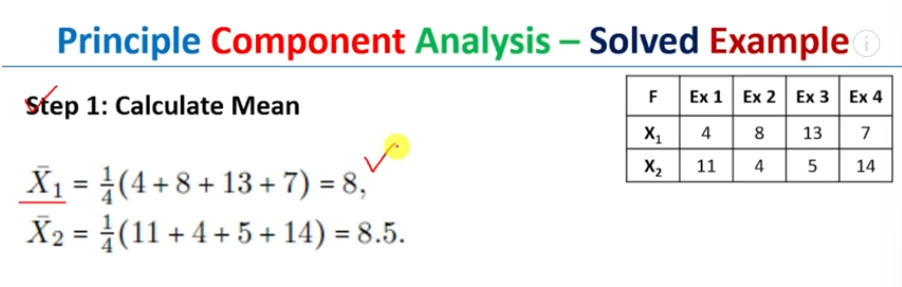
The number or scalar value λ is an eigenvalue of A. Eigenvector corresponds to the real non zero eigenvalues which point in the direction stretched by the transformation whereas eigenvalue is considered as a factor by which it is stretched. In case, If the eigen value is negative, the direction of the transformation is negative.

Suppose is called an Eigen or characteristic matrix, which is an indefinite and undefined scalar. Where determinant of eigen matrix can be written as is the eigen equation or characteristic equation, where I is the identity matrix.

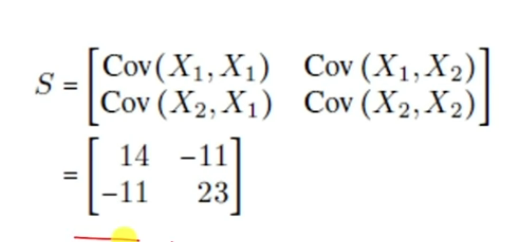
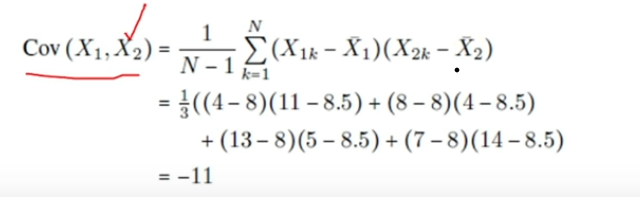
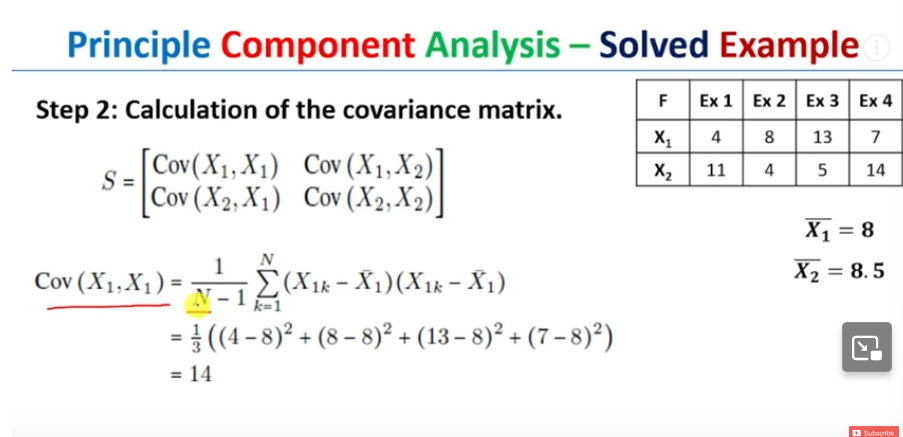
Eigen vectors: Eigen vectors are the vectors (non-zero) that do not change the direction when any linear transformation is applied. It changes by only a scalar factor. In a brief, we can say, if A is a linear transformation from a vector space V and x is a vector in V which is not a zero vector, then v is an eigenvector of A if A(x) is a scalar multiple of x.

Example:

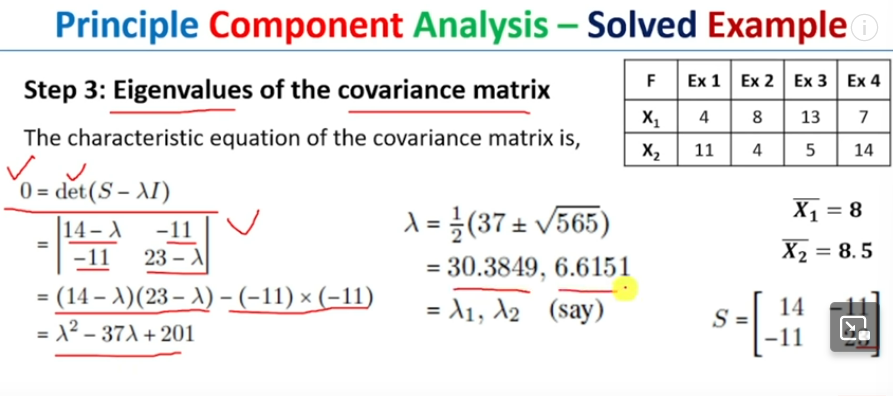
Let’s assume we have a below dataset with two feature and we have to reduce the dimensionality of it. First, we will evaluate the mean of the respective features.



Once we calculate the mean then we will calculate the covariance as below



Once we get the co variance matrix, we have to calculate the eigen value of the co variance matrix



Compute Eigenvectors:

