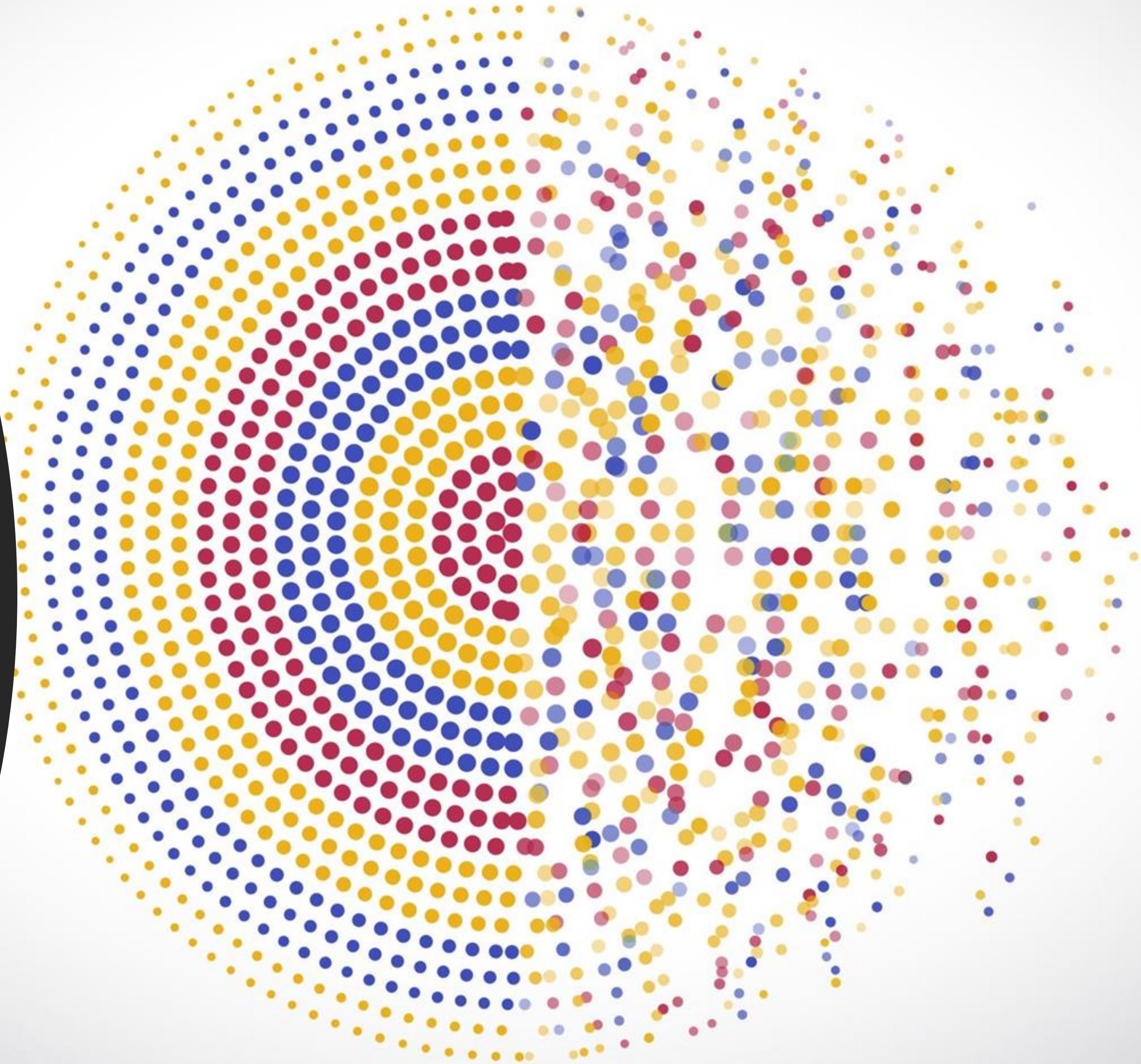
# Data Preprocessing

Dr. Mohamed AlHajri



# Data Preprocessing

- Dimensionality Reduction
- Feature Selection
- Feature Creation

# Dimensionality Reduction

- Dimensionality reduction  is a technique that transforms high-dimensional data into a lower-dimensional space:
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
  - UMAP (Uniform Manifold Approximation and Projection)
  - Random Projection

# Principal Component Analysis

- PCA is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while capturing the directions of maximum variance. This is widely used for tasks such as data compression, noise reduction, and visualization.

# Principal Component Analysis

- Before performing PCA, we center the data by subtracting the mean of each feature. This ensures that the principal components align with the directions of maximum variance, and that the covariance matrix accurately represents the relationships between features.
- For a dataset $X \in R^{n\times d}$, where n is the number of samples and d is the number of features, we compute the mean vector $\mu \in R^{1\times d}$ for each feature. Subtracting the mean results in the centered data matrix $X_{centered} \in R^{n\times d}$:

$$\boxed{X_{centered} = X - \mu}$$

- The covariance matrix $\Sigma \in R^{d\times d}$ is then computed as:

$$\boxed{\Sigma = \frac{1}{n-1} X_{centered}^{T} X_{centered}}$$

Each element $\Sigma_{ij}$ represents the covariance between features $i$ and $j$, with the diagonal elements representing the variance of individual features.

# Principal Component Analysis

- **Eigenvalue Decomposition**
- PCA finds the principal components by solving the eigenvalue decomposition of the covariance matrix:

$$\boxed{\Sigma\, v_i \;=\; \lambda_i v_i}$$

Where:

- $v_i \;\in\; R^{d\times 1}$ are the eigenvectors (principal components).
- $\lambda_i \;\in\; R$ are the eigenvalues, representing the variance captured by each component.

- The top k eigenvectors form the matrix $W\_k \;\in\; R^{d\times k}$, which projects the original data into a lower-dimensional subspace:

$$\boxed{Y \;=\; X_{centered} W_k,} \qquad where\; Y \in R^{n \times k}.$$

# Principal Component Analysis

- **Singular Value Decomposition (SVD)**
- Alternatively, PCA can be performed using Singular Value Decomposition (SVD). SVD decomposes the original data matrix $X \in R^{n\times d}$ as:

$$X = U \Sigma V^T$$

Where:

- $U \in R^{\{n \times n\}}$ contains the left singular vectors.
- $V \in R^{\{d \times d\}}$ contains the right singular vectors (the principal components).
- $\Sigma \in R^{\{n \times d\}}$ contains the singular values.

# Principal Component Analysis

- **Connection Between Maximizing Variance and Minimizing Error**
- The objective of PCA is to find a projection matrix $W\_k \in R^{\{d \times k\}}$ that maximizes the variance of the projected data. The variance of the projected data $Y = X_{\{centered\}} W_k$ can be expressed as:

$$Var(Y) = \frac{1}{n-1} Y^T Y = \frac{1}{n-1} W_k^T X_{\{centered\}}^T X_{\{centered\}} W_k = \underset{\substack{\downarrow \\ \text{covariance} \\ \text{of original matrix}}}{W_k^T \Sigma W_k}$$

To maximize the variance, we solve the optimization problem:

$$\boxed{max_W \; tr(W^T \Sigma W)}, subject\ to\ W^T W = I.$$

# Principal Component Analysis

- **Connection Between Maximizing Variance and Minimizing Error**
- Minimizing the reconstruction error is an alternative interpretation of PCA. After projecting the data onto the lower-dimensional subspace, we attempt to reconstruct the original data, minimizing the Frobenius norm of the difference between the original data and its reconstruction:

$$\boxed{min_W \ \left|\left|X_{\{centered\}} - \underbrace{X_{\{centered\}} \ W}_{Y \in \mathbb{R}^{n \times k}} \ \underbrace{\overset{\text{mapping matrix}}{W^T}}_{W \in \mathbb{R}^{d \times k}}\right|\right|_F^2 .}$$

$\Rightarrow W^T \in \mathbb{R}^{k \times d}$

The expanded Frobenius norm is:

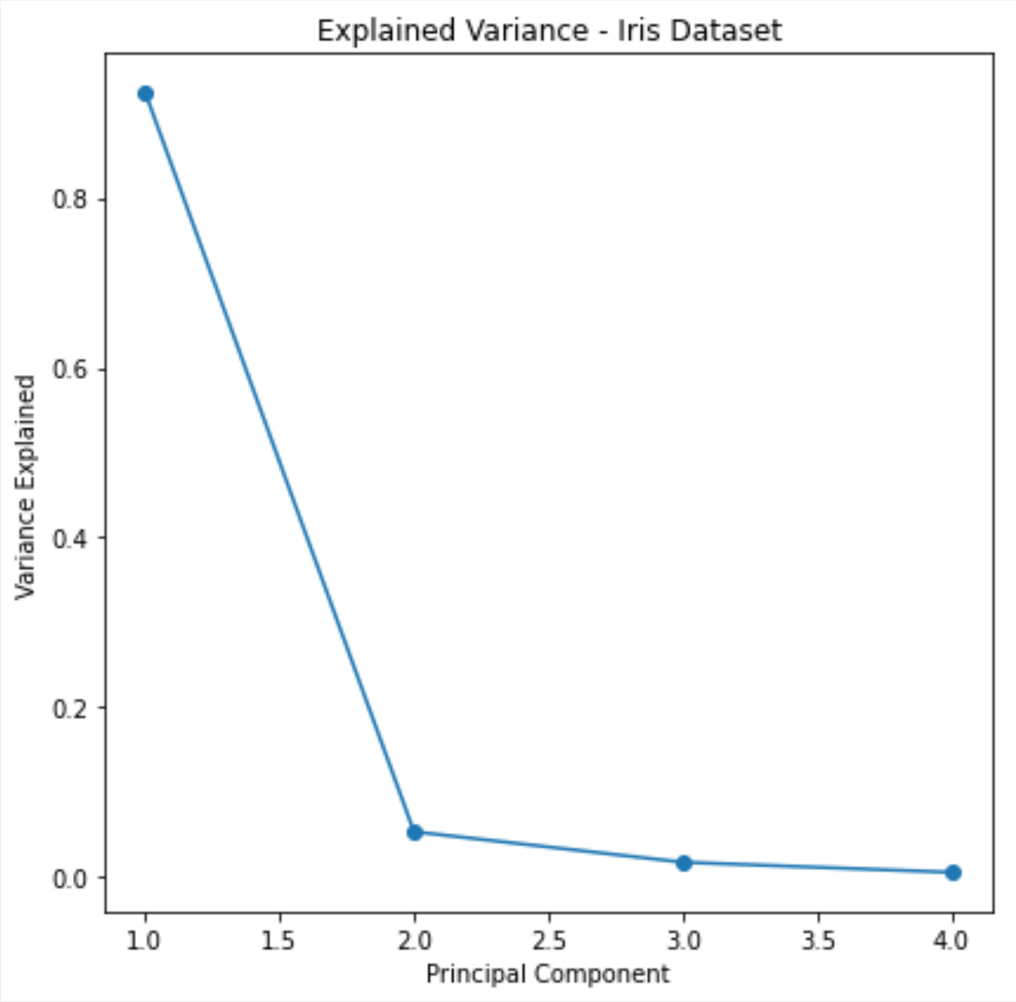
$: Y \cdot W^T = X_{reconstructed} \in \mathbb{R}^{n \times d}$

$$||X_{\{centered\}} - X_{\{centered\}} \ W \ W^T \ ||_F^2$$
$$= tr(X_{\{centered\}}^T \ X_{\{centered\}}) - 2tr(W^T \ X_{\{centered\}}^T \ X_{\{centered\}} \ W)$$
$$+ \ tr(W^T \ X_{\{centered\}}^T \ X_{\{centered\}} \ W \ W^T \ W).$$

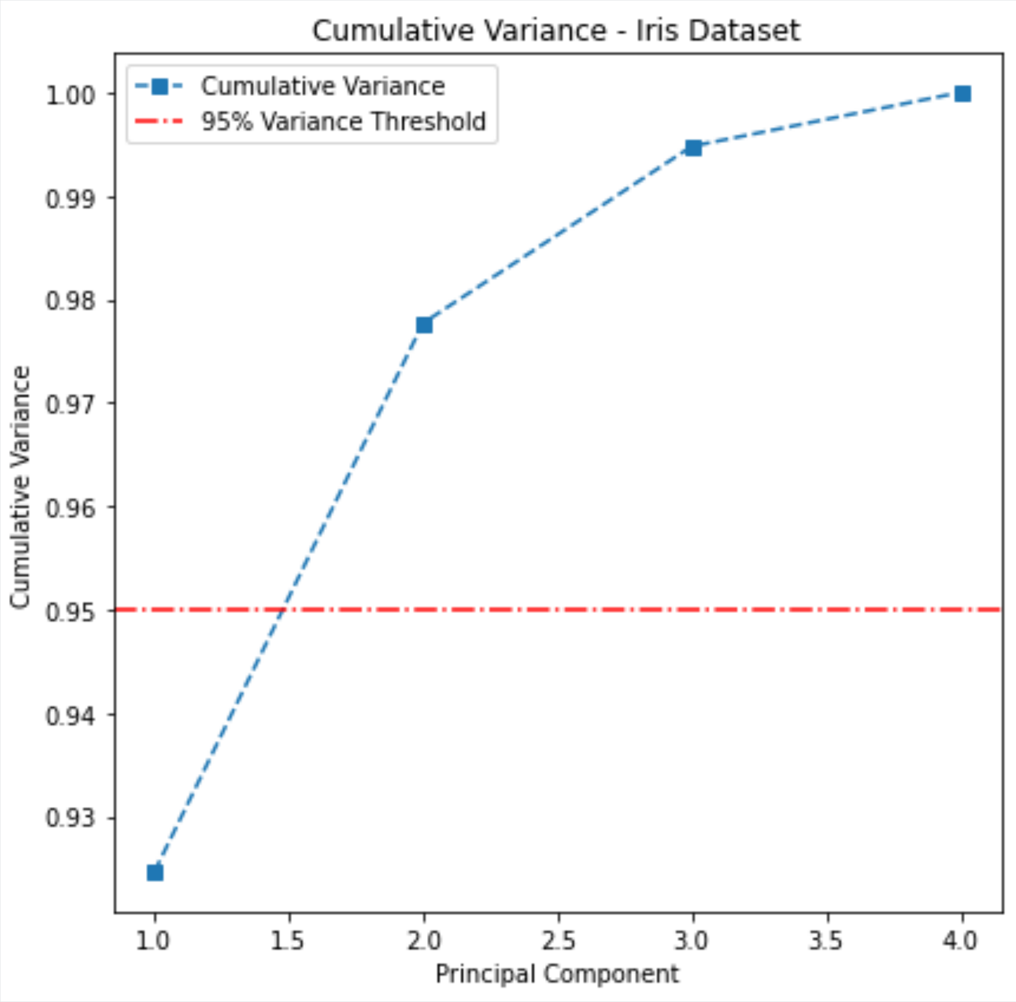
# Principal Component Analysis

- **Choosing the Number of Principal Components**
- The following methods help to determine the optimal number of principal components:
  1. Elbow Method.
  2. Cumulative Explained Variance.
  3. Cross-Validation.

# Principal Component Analysis

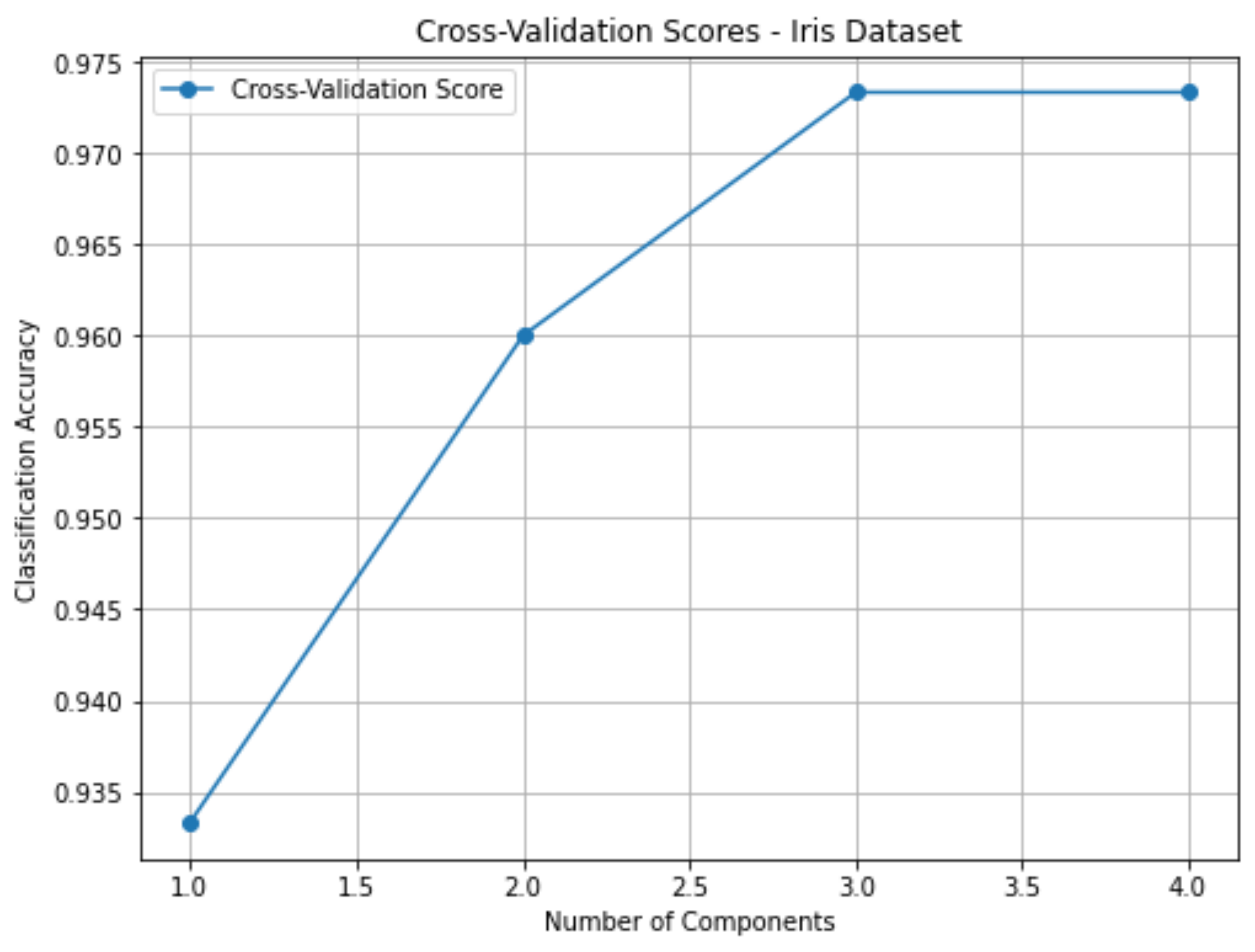


Elbow Method



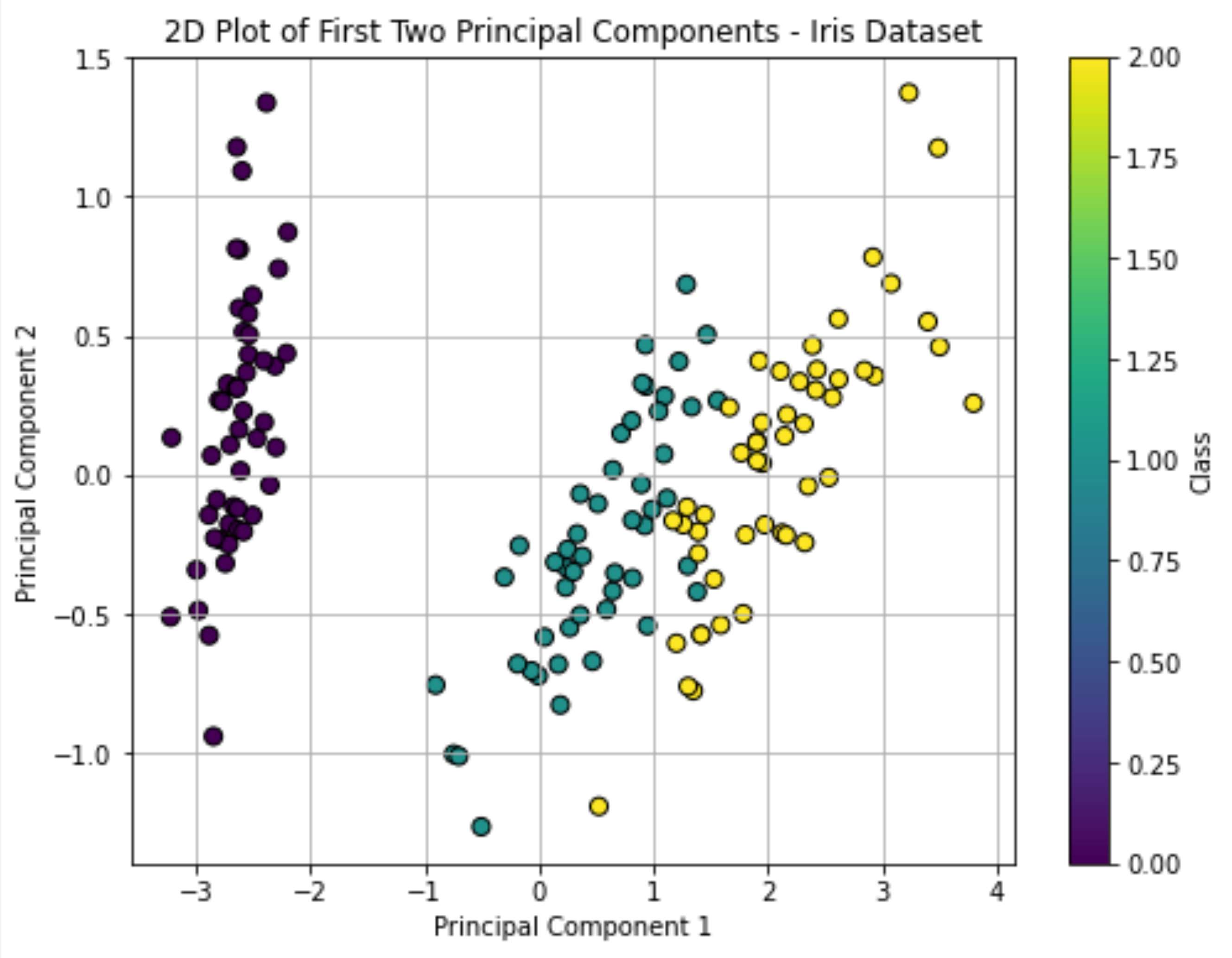
Cumulative Explained Variance.

# Principal Component Analysis

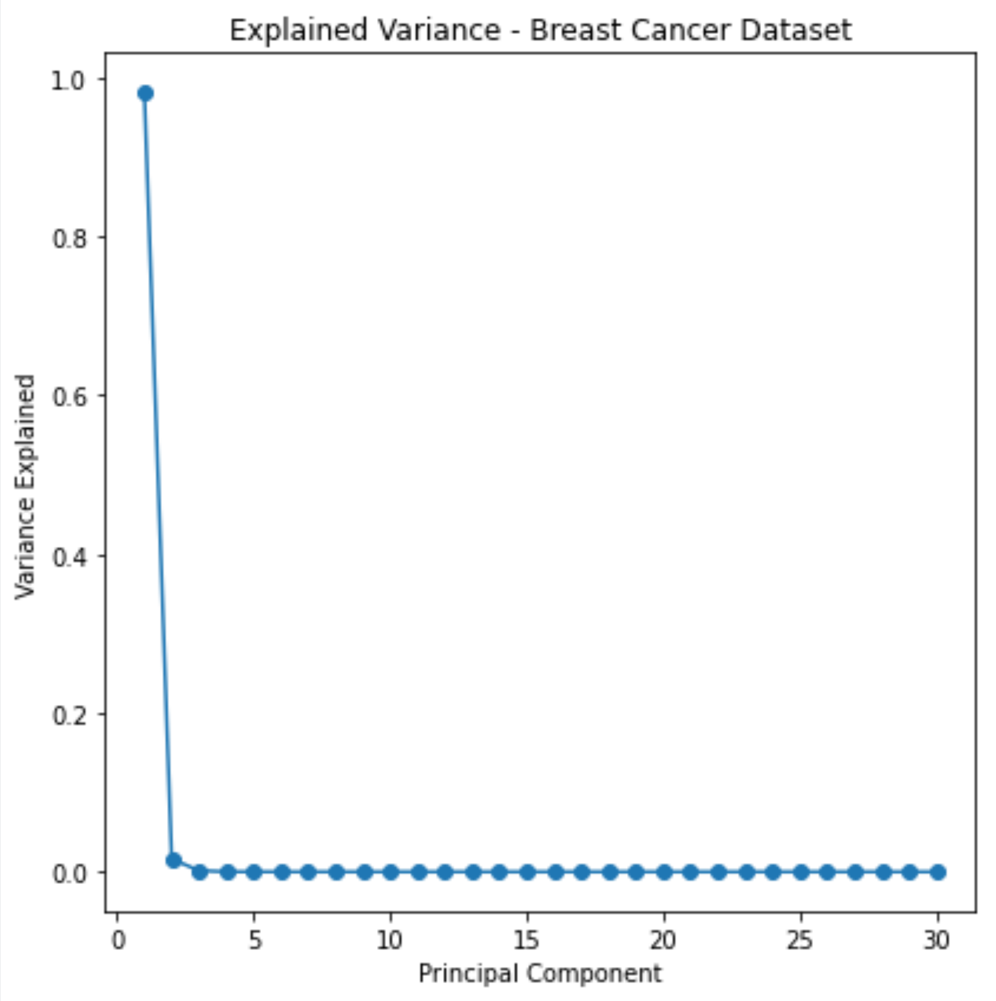


Cross Validation

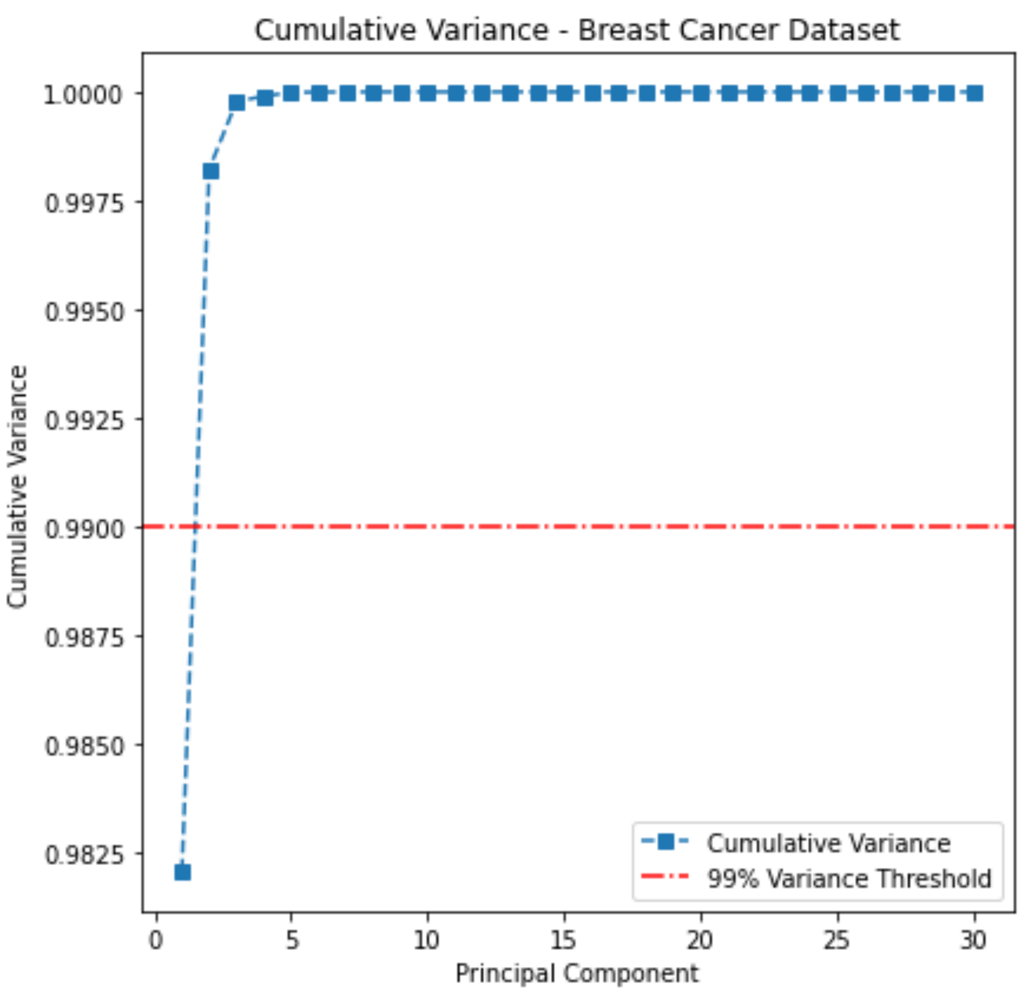
# Principal Component Analysis



# Principal Component Analysis

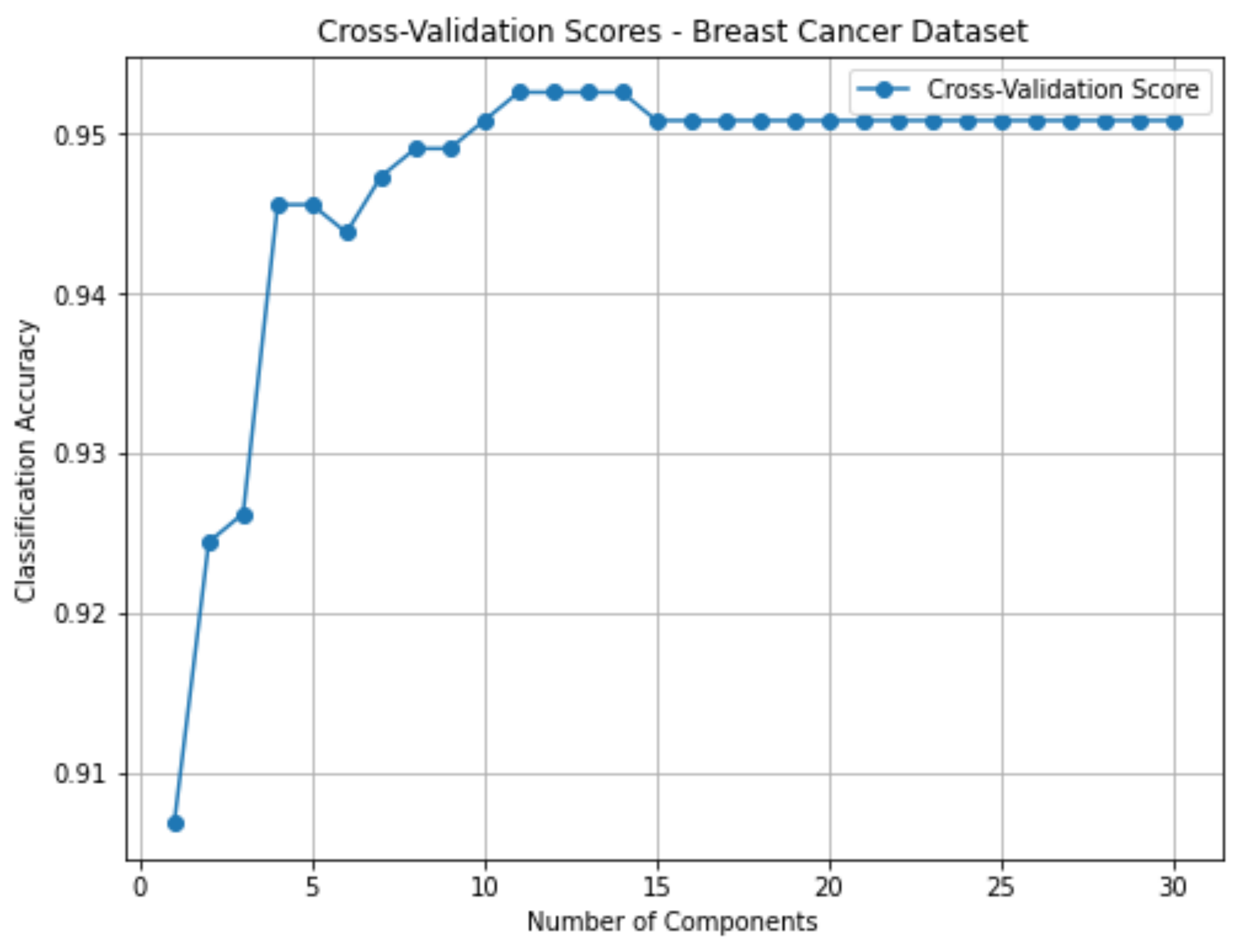


Elbow Method



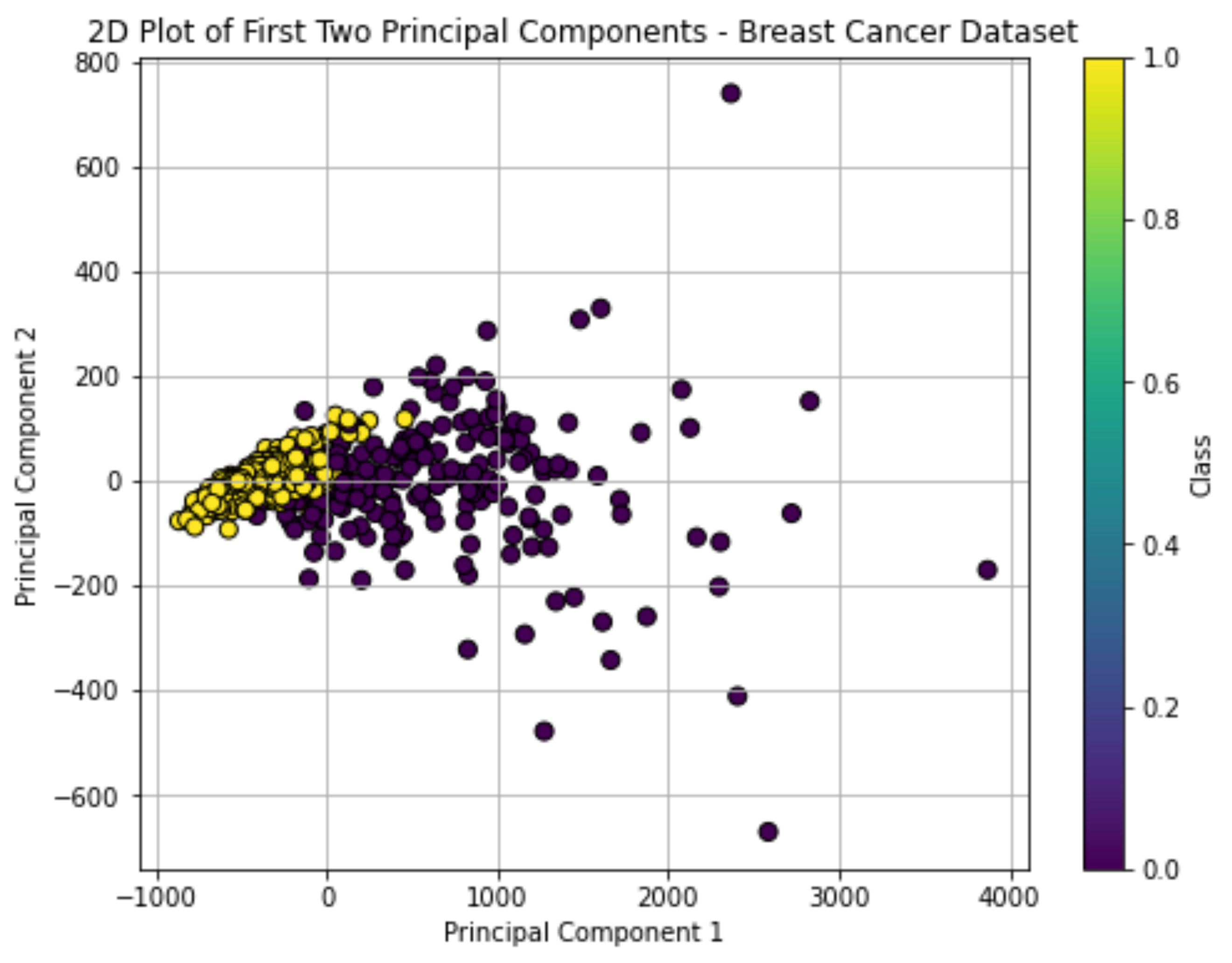
Cumulative Explained Variance.

# Principal Component Analysis



Cross Validation

# Principal Component Analysis



# Principal Component Analysis

- **Limitations**
- Linearity: PCA assumes that the principal components are linear combinations of the original features. It fails to capture non-linear relationships in the data.
- Interpretability: The principal components are often linear combinations of many features, making it difficult to interpret their meaning.
- Variance-Based: PCA focuses on maximizing variance, which may not always capture the most important features for specific tasks such as classification.
- Sensitivity to Scaling: PCA is sensitive to the scaling of features. Features with higher magnitude may dominate the principal components unless the data is appropriately scaled.

# Feature Selection

- Feature selection is the process of selecting the most relevant features from a dataset to improve the performance of a machine learning model. It helps by reducing the dimensionality of the feature space, improving model accuracy, and decreasing computational costs. There are three main types of feature selection methods: **filter methods**, **wrapper methods**, and **embedded methods**.

# Feature Selection

- **Filter Methods**
- Filter methods are statistical techniques that rank and select features based on their properties, independent of any specific machine learning model. They are fast and efficient, making them suitable for pre-processing steps.
    - Variance Threshold
    - Correlation Coefficient
    - Mutual Information

# Feature Selection

**Filter Method - Variance Threshold**

- Variance threshold removes features that have a variance below a specified threshold. Features with low variance typically carry little information.
- **Mathematics**: For a feature $\boldsymbol{X}_j \in R^n$, the variance is:

$$\boxed{Var(X_j) = \frac{1}{n}\sum(X_{ij} - \mu_j)^2}$$

Features with variance below the threshold are removed.

# Feature Selection

## Filter Method - Correlation Coefficient

- The Pearson correlation coefficient measures the linear relationship between a feature and the target variable. Features with low correlation to the target are considered irrelevant.
- **Mathematics**: The Pearson correlation coefficient between a feature $X_j$ and target $y$ is:

$$\boxed{\rho(X_j, y) = \frac{Cov(X_j, y)}{\sigma_X \sigma_y}}$$

is the covariance between $X_j$ and $y$, and $\sigma_{X_j}$and $\sigma_y$ are the standard deviations.

# Feature Selection

## Filter Method - Mutual Information

- Mutual information (MI) measures the dependency between a feature and the target variable. It captures both linear and non-linear relationships.
- **Mathematics**: The mutual information between a feature $X_j$ and target $y$ is:

$$I(X_j; y) = \sum_{x \in X_j, \alpha \in y} p(x, \alpha) \log(\frac{p(x, \alpha)}{p(x)p(\alpha)})$$

Higher MI indicates a stronger relationship. $= 0 \Rightarrow$ not linearly correlated

# Feature Selection

- **Wrapper Methods**
- Wrapper methods evaluate subsets of features using a machine learning model to determine which features contribute the most to model performance. These methods are computationally expensive but often more accurate than filter methods.
    - Recursive Feature Elimination
    - Forward Feature Selection

# Feature Selection

**Wrapper Method – Recursive Feature Elimination**

- Recursive Feature Elimination (RFE) ranks features based on their importance and recursively eliminates the least important ones.
- **Process**:
  - Train a model with all features.
  - Rank the features based on their importance (e.g., model coefficients).
  - Remove the least important feature.
  - Repeat until the desired number of features remains.

# Feature Selection

**Wrapper Method – Forward Feature Selection**

- Forward selection starts with no features and adds features one by one, selecting the feature that improves model performance the most at each step.

# Feature Selection

## Embedded Methods

- Embedded methods perform feature selection during the model training process. Lasso regression is one of the most common embedded methods, as it automatically selects features by penalizing the coefficients of less important features.
- **Lasso Regression (L1 Regularization)**
- Lasso regression adds an $L_1$ regularization term to the objective function, which forces some feature coefficients to become zero, effectively selecting a subset of features.
- **Mathematics**: Lasso regression solves the following optimization problem:
- $\min\limits_{w} \frac{1}{2n} || Xw - y ||_2^2 + \lambda || w ||_1$
- where λ controls the strength of regularization. Features corresponding to zero coefficients are removed.