

**Unit II**  
**(BE Computer 2019 PAT)**  
**A.Y. 2025-26**

**By Payal Hake**

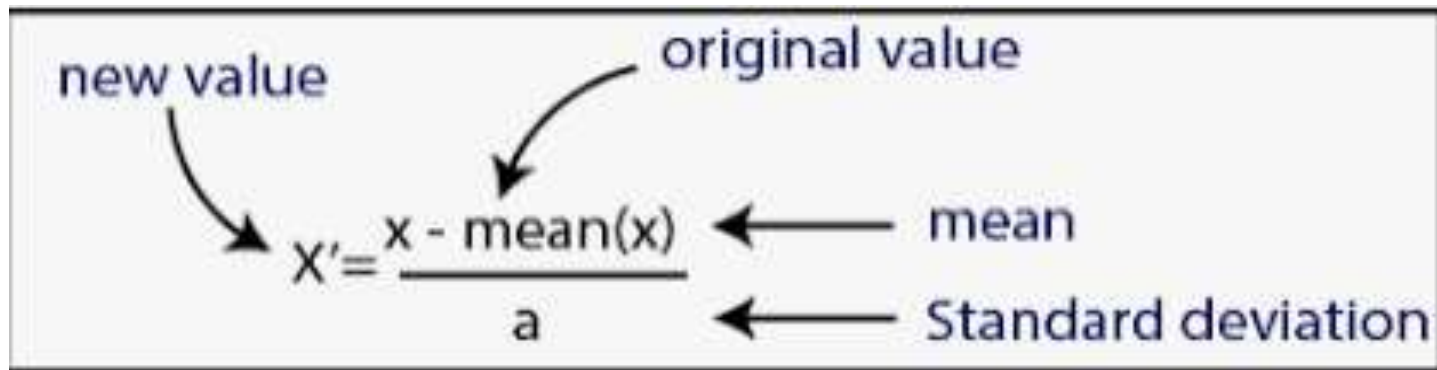
# Unit-2 Feature Engineering

- Concept of Feature, Preprocessing of data: Normalization and Scaling, Standardization, Managing 1Hr
- missing values, Introduction to Dimensionality Reduction, Principal Component Analysis (PCA), 1Hr
- Feature Extraction: Kernel PCA, Local Binary Pattern. 1Hr
- Introduction to various Feature Selection Techniques, Sequential 1Hr
- Forward Selection, Sequential Backward Selection. 1Hr
- Statistical feature engineering: count-based, Length, Mean, Median, Mode etc. based feature vectorcreation. 1Hr
- Multidimensional Scaling, Matrix Factorization Techniques. 1Hr

# Introduction to Feature Preprocessing

## What is Feature Preprocessing?

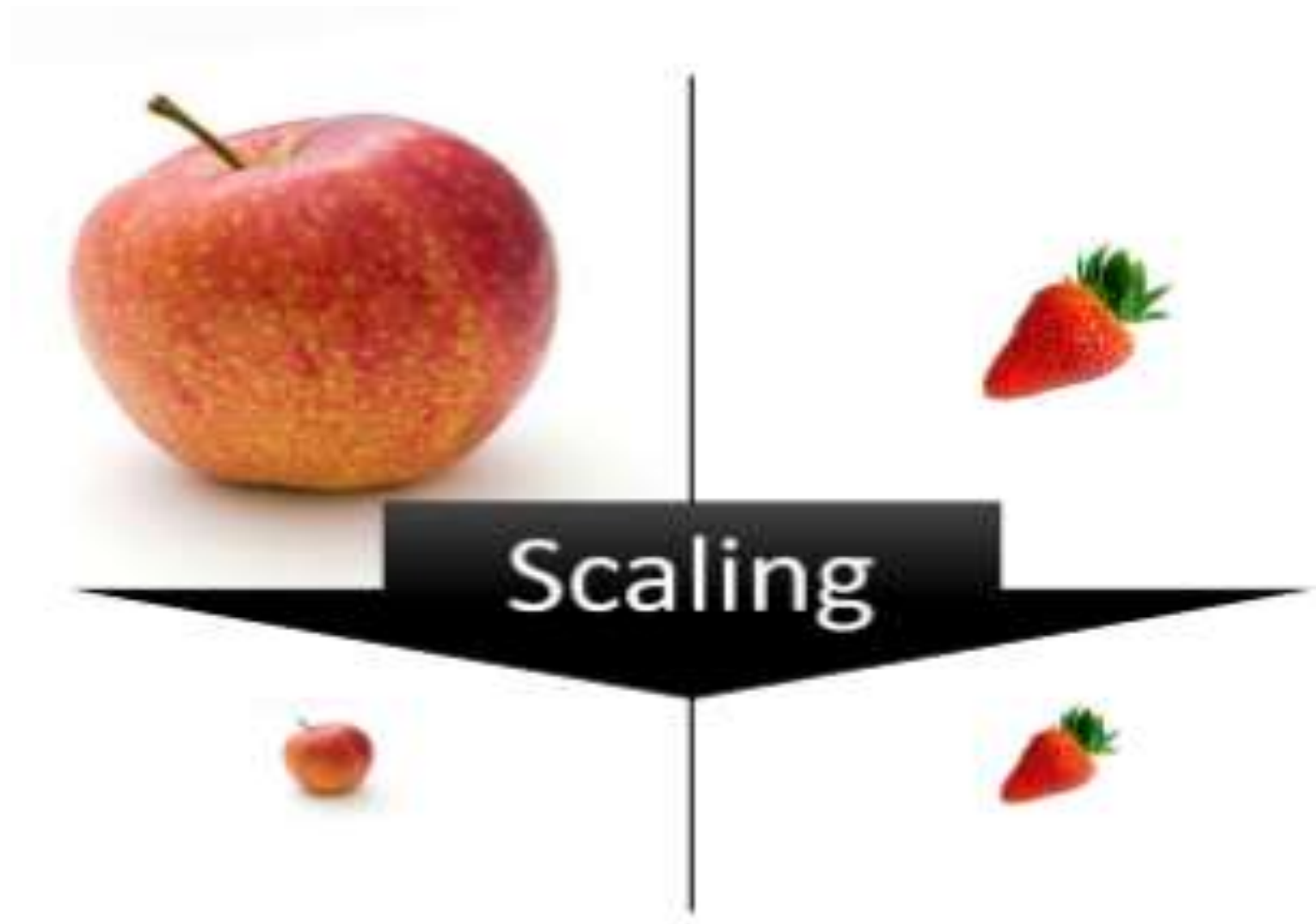
- It is a crucial step in the machine learning workflow before model training.
- It involves transforming raw data into a suitable format for the model.
- This process can significantly impact the performance, accuracy, and efficiency of your machine learning model.



The diagram illustrates the formula for z-score normalization, which is a common feature preprocessing technique. The formula is presented as 
$$X' = \frac{x - \text{mean}(x)}{a}$$
 within a rectangular box. An arrow labeled "new value" points to the variable  $X'$  on the left side of the equation. Another arrow labeled "original value" points to the variable  $x$  in the numerator. A horizontal arrow labeled "mean" points from the text "mean" to the  $\text{mean}(x)$  term in the numerator. A second horizontal arrow labeled "Standard deviation" points from the text "Standard deviation" to the variable  $a$  in the denominator.

# Introduction to Feature Preprocessing

- **Why is it Essential?**
  - **Data Inconsistencies:** Raw data often contains noise, inconsistencies, and missing values.
  - **Algorithm Requirements:** Many machine learning algorithms have specific data requirements (e.g., features must be centered at 0 or scaled to a certain range).
  - **Improving Model Performance:** Proper preprocessing can lead to faster convergence, better model accuracy, and more robust predictions.
- **Analogy:** Think of feature preprocessing as preparing ingredients for a recipe. The final dish's quality depends heavily on how well the ingredients are prepared.



**Apple & Strawberry**

# The Importance of Feature Scaling

## What is Feature Scaling?

- Feature scaling is a method used to **normalize the range of independent variables or features of data.**
- It ensures **all features contribute equally to the model**, preventing a feature with a larger magnitude from dominating others.
- If we want to get the best-mixed juice, we need to mix all fruit not by their size but based on their right proportion.
- We just need to remember apple and strawberry are not the same unless **we make them similar in some context to compare their attribute.**
- Similarly, in many **machine learning algorithms**, to bring all features in the same standing, we need to do scaling so that **one significant number doesn't impact the model** just because of their large magnitude.

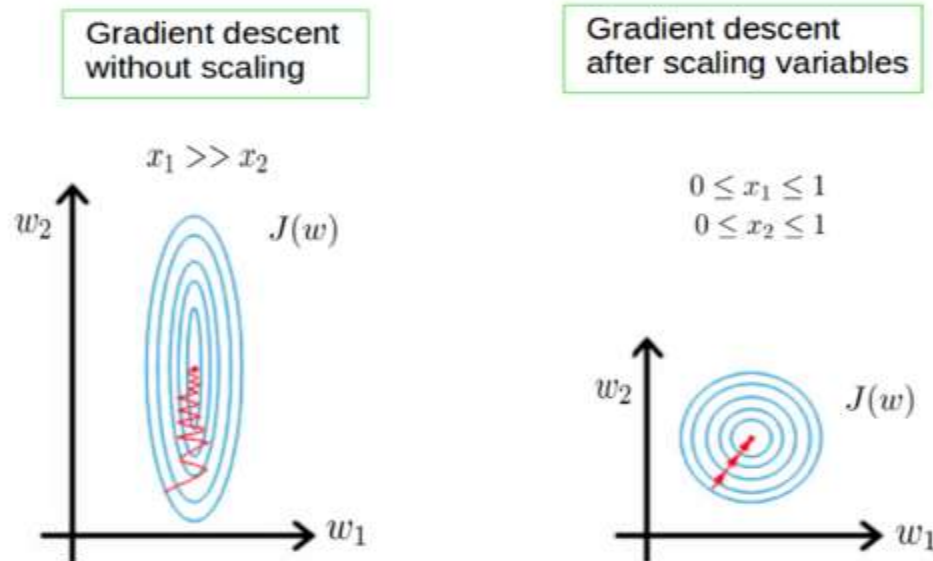
# The Importance of Feature Scaling:

## When and Why to Use It?

- **Distance-Based Algorithms:** Algorithms like K-Nearest Neighbors (KNN) and K-Means Clustering use distance measures (e.g., Euclidean distance). Without scaling, features with larger magnitudes will disproportionately influence the distance calculation.
- **Gradient Descent-Based Algorithms:** Algorithms like Linear Regression, Logistic Regression, and Neural Networks use gradient descent. Scaling can speed up the convergence of the optimization algorithm.
- **Principal Component Analysis (PCA):** PCA is sensitive to the variance of features. Features with high variance (often due to large magnitude) can skew the principal components, leading to a biased analysis. Scaling ensures each feature's variance is on the same footing.
- **Support Vector Machines (SVMs):** Scaling helps find support vectors faster and more efficiently.
- **Key Algorithms Impacted:** K-Nearest Neighbors (KNN), K-Means Clustering, Principal Component Analysis (PCA), Neural Networks.

# The Importance of Feature Scaling:

- Feature scaling is needed to bring every feature in the same footing without any upfront importance.
- if we convert the weight to “Kg,” then “Price” becomes dominant.
- **E.g.** Neural network gradient descent **converge much faster** with feature scaling than without it.





## Feature Scaling Techniques: Normalization vs. Standardization

- **Feature scaling** in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model.
- **Scaling can make a difference between a weak machine learning model and a better one.**

**Most common techniques of feature scaling are**

- **Normalization and Standardization.**

# Normalization (Min-Max Scaling)

- Objective: To scale feature values to a fixed range, typically [0, 1].

Formula:  $X_{normalized} = \frac{(X - X_{min})}{(X_{max} - X_{min})}$ .

- **Xn** = Value of Normalization
- **Xmaximum** = Maximum value of a feature
- **Xminimum** = Minimum value of a feature

## Characteristics:

- Bounds values between a minimum and maximum.
- Does not center the mean at 0.
- **Sensitive to Outliers:** Outliers can disproportionately affect the maximum and minimum values, squashing the range of other data points.

**When to use:** When the feature distribution is not Gaussian and for algorithms that require features to be within a specific range (e.g., some neural networks).

# Normalization

**Case1-** If the value of X is minimum, the value of Numerator will be 0; hence Normalization will also be 0.

$$X_n = (X - X_{\text{minimum}}) / (X_{\text{maximum}} - X_{\text{minimum}})$$

Put  $X = X_{\text{minimum}}$  in above formula, we get;

$$X_n = X_{\text{minimum}} - X_{\text{minimum}} / (X_{\text{maximum}} - X_{\text{minimum}})$$

$$X_n = 0$$

**Case2-** If the value of X is maximum, then the value of the numerator is equal to the denominator; hence Normalization will be 1.

$$X_n = (X - X_{\text{minimum}}) / (X_{\text{maximum}} - X_{\text{minimum}})$$

Put  $X = X_{\text{maximum}}$  in above formula, we get;

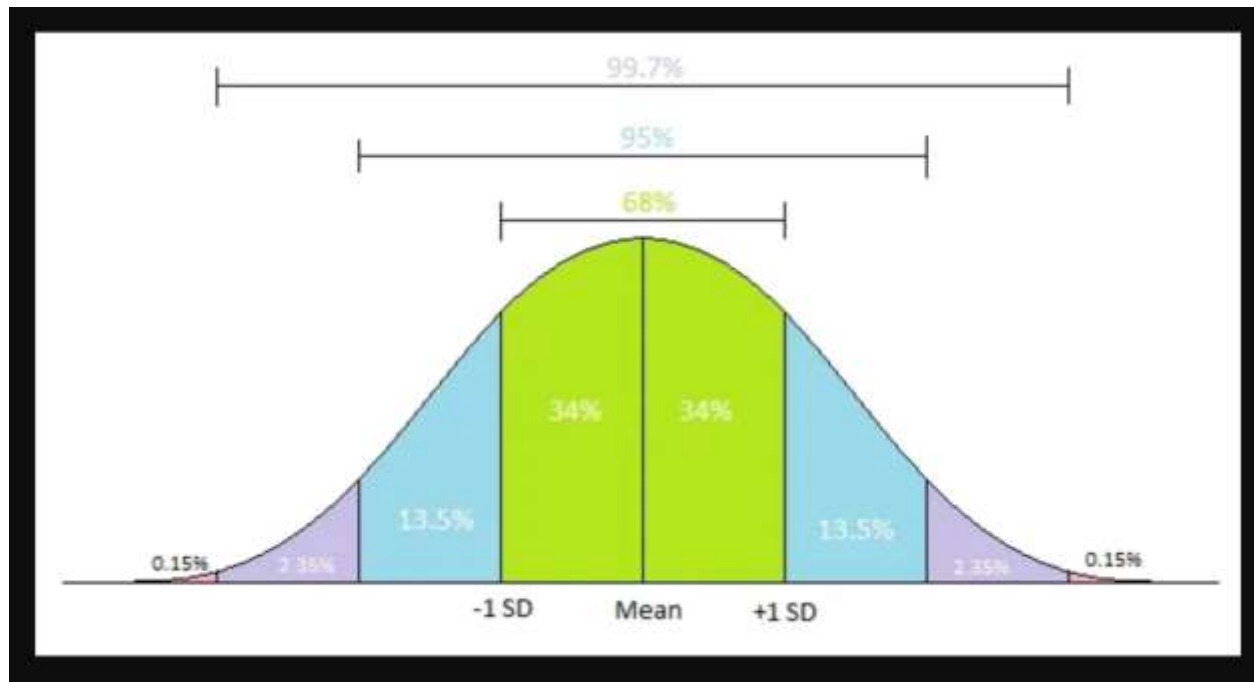
$$X_n = X_{\text{maximum}} - X_{\text{minimum}} / (X_{\text{maximum}} - X_{\text{minimum}})$$

# Normalization

- **Case3-** On the other hand, if the value of **X** is neither maximum nor minimum, then values of normalization will also be between 0 and 1.
- Hence, Normalization can be defined as a scaling method where values are shifted and rescaled to maintain their ranges between 0 and 1, or in other words; it can be referred to as **Min-Max scaling technique**.
- Normalization techniques in Machine Learning
- **Min-Max Scaling:** This technique is also referred to as scaling. ranging between 0 and 1.

# Standard Deviation ( $\sigma$ )

- A measure of the **amount of variation or dispersion** of a set of values.
- A **low standard deviation** indicates that the **values tend to be close to the mean**, while a **high standard deviation** indicates that the values are **spread out over a wider range**.
- It is the **square root of the variance**.
- It is denoted by the lower Greek letter  $\sigma$  (**sigma**).

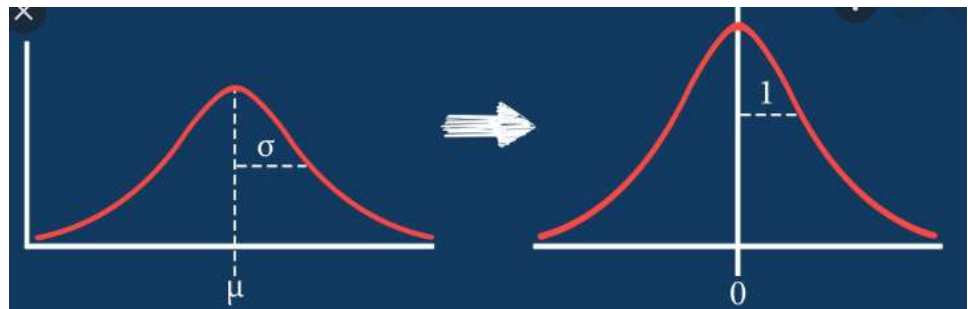


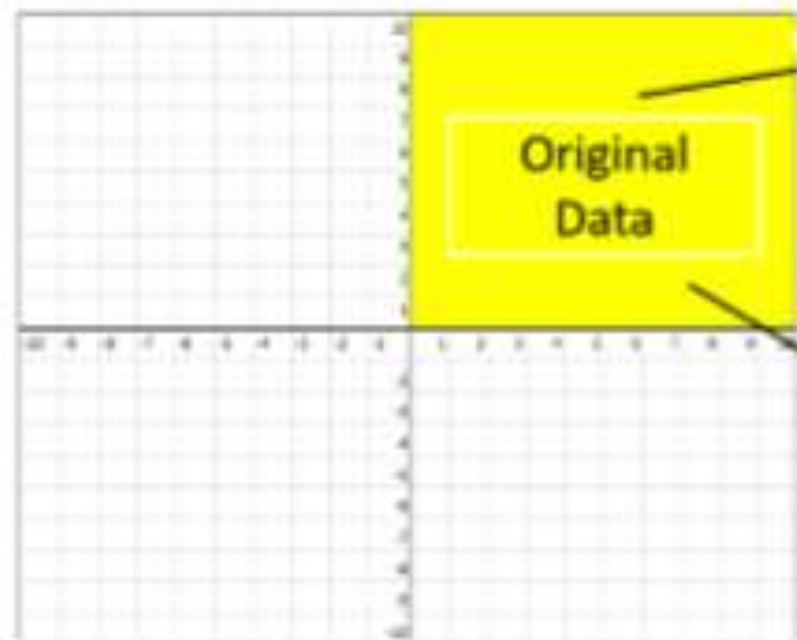
# Standardization (Z-Score Scaling)

- **Objective:** To transform data to have a **mean of 0** and a **standard deviation of 1**.

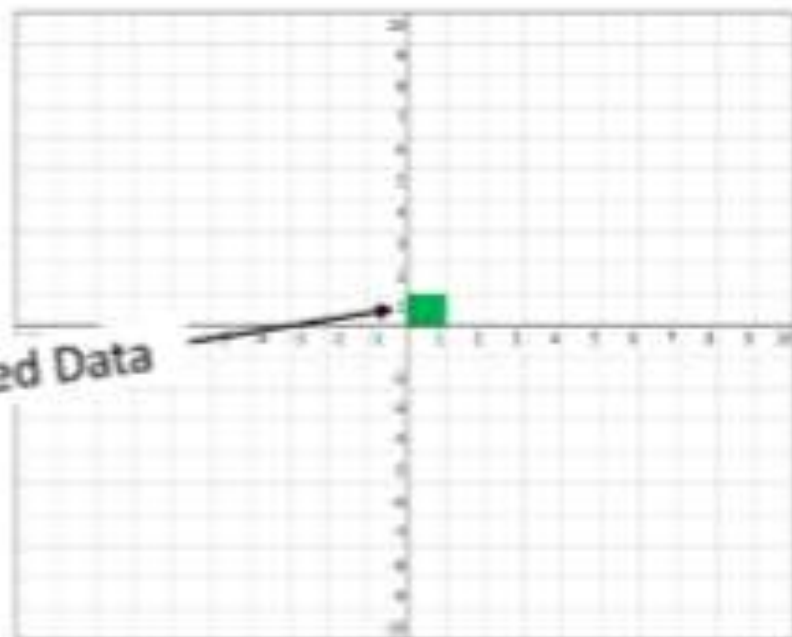
**Formula:**  $Z = \frac{(X - \mu)}{\sigma}$  (where  $\mu$  is the mean and  $\sigma$  is the standard deviation).

- **Characteristics:**
- Centers the mean at 0 and scales the variance at 1.
- **Preserves Outliers:** It is not bounded, so outliers are not compressed.
- Preserves the shape of the original distribution.
- **When to use:** When the data follows a Gaussian distribution (bell curve) and for algorithms like Linear Regression, Logistic Regression, and SVMs.

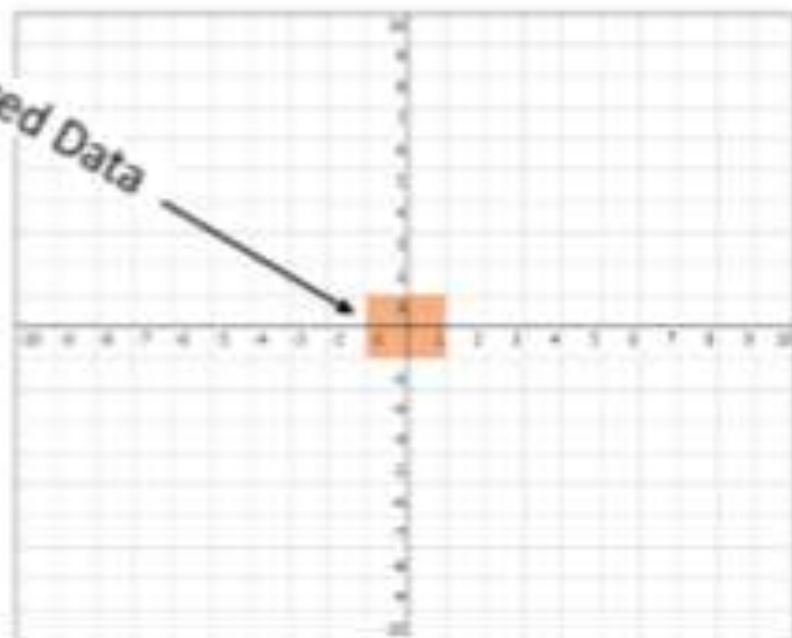




Normalized Data



Standardized Data



## Concept of Feature Preprocessing of data:

- Suppose we have two features of weight and price, as in the below table. The “Weight” cannot have a meaningful comparison with the “Price.” **So the assumption algorithm makes that since “Weight” > “Price,” thus “Weight,” is more important than “Price.”**

Name	Weight	Price
Orange	15	1
Apple	18	3
Banana	12	2
Grape	10	5



# Variance ( $\sigma^2$ )

- The average of the squared differences from the mean.
- It measures how much a random variable differs from its expected value.
- It is always a non-negative value.

## Example:

**Data Set A:** [1, 2, 3, 4, 5] → Low Variance, Low Standard Deviation.

**Data Set B:** [1, 10, 20, 30, 40] → High Variance, High Standard Deviation.

# Difference between Normalization and Standardization

Normalization	Standardization
This technique uses minimum and max values for scaling of model.	This technique uses mean and standard deviation for scaling of model.
It is helpful when features are of different scales.	It is helpful when the mean of a variable is set to 0 and the standard deviation is set to 1.
Scales values ranges between [0, 1] or [-1, 1].	Scale values are not restricted to a specific range.
It got affected by outliers.	It is comparatively less affected by outliers.
Scikit-Learn provides a transformer called MinMaxScaler for Normalization.	Scikit-Learn provides a transformer called StandardScaler for Normalization.
It is also called Scaling normalization.	It is known as Z-score normalization.
It is useful when feature distribution is unknown.	It is useful when feature distribution is normal.

# Missing Data Values And How To Handle It



# Handling Missing Data Values

**Missing Data in Real-World Scenarios:** Datasets often have missing values due to human error, equipment failure, or data collection issues.

## Three Main Types:

- **Missing Completely At Random (MCAR):** Missingness is independent of both observed and unobserved data. **Example:** A librarian randomly forgets to enter data.
- **Missing At Random (MAR):** Missingness depends on observed data but not on the missing data itself. **Example:** Men are less likely to respond to a survey, so missingness is related to the gender variable.
- **Missing Not At Random (MNAR):** Missingness is dependent on the value of the missing data itself. **Example:** People with high incomes are less likely to report their salary.

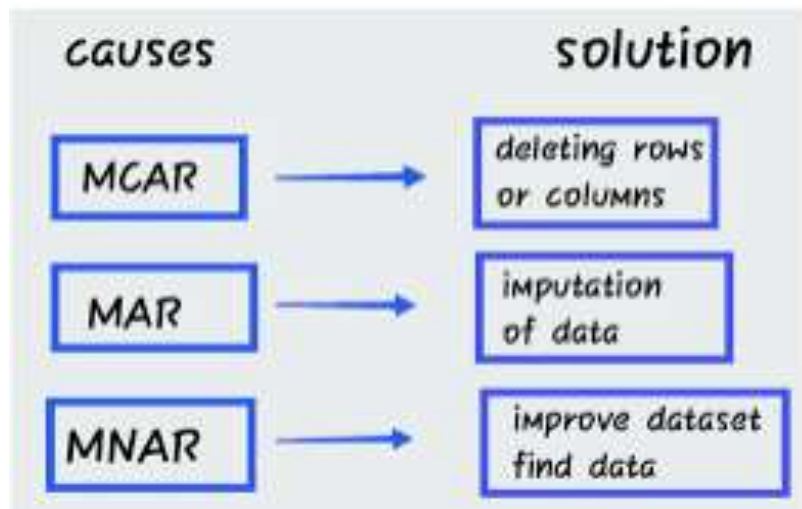
# Strategies for Handling Missing Data:

## Deletion:

- **Listwise Deletion:** Removing entire rows with missing values. Simple but can lead to significant data loss.
- **Pairwise Deletion:** Using all available data for a specific analysis, ignoring missing values.

## Imputation:

- **Simple Imputation:** Replacing missing values with a single constant like the mean, median, or mode.
- **Arbitrary Value Imputation:** Replacing missing values with a placeholder value (e.g., -99).
- **Advanced Imputation:** Using more sophisticated methods like K-Nearest Neighbors (KNN) or machine learning models to predict the missing values.



# Introduction to Dimensionality Reduction

## What is Dimensionality?

The number of input features, variables, or columns in a dataset.

## What is Dimensionality Reduction?

- The number of input features, variables, or columns present in a given dataset is known as **Dimensionality**, and the process to reduce these features is called **dimensionality reduction**.
- *"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."*

# Introduction to Dimensionality Reduction

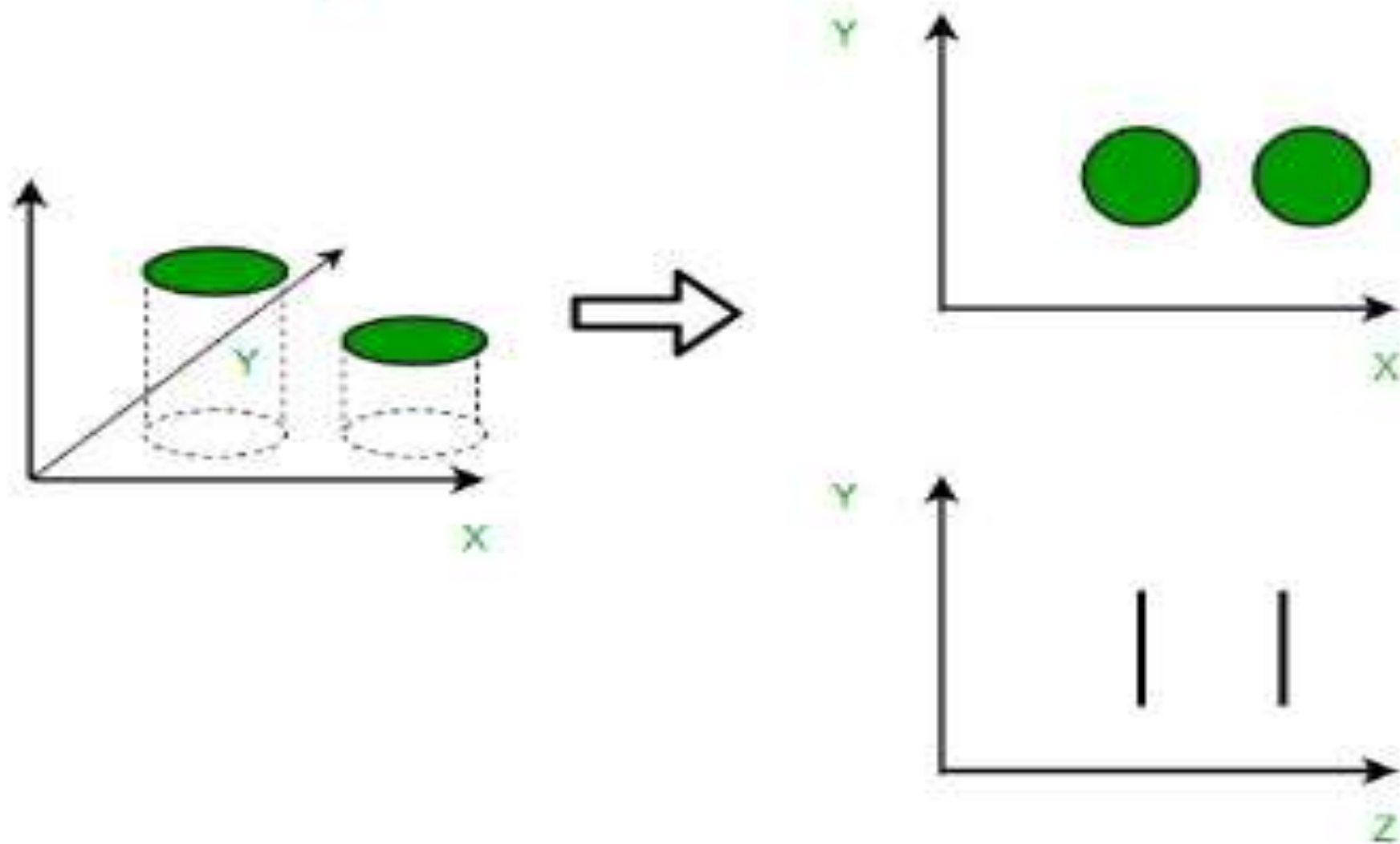
## Why is it Important?

- **Curse of Dimensionality:** In high-dimensional spaces, data becomes sparse, making it difficult for models to find meaningful patterns.
- **Reduced Storage Space:** A dataset with fewer dimensions requires less storage.
- **Faster Training:** Less computation is needed to train a model with fewer features.
- **Data Visualization:** Easier to visualize data in 2D or 3D spaces.
- **Removes Redundant Features:** Helps in handling multicollinearity by removing correlated features.

## Components of Dimensionality Reduction:

- **Feature Selection:** Choosing a subset of the most relevant features from the original dataset.
- **Feature Extraction:** Creating new, lower-dimensional features from the original high-dimensional features.

## Dimensionality Reduction



**Components of Dimensionality Reduction**



# Dimensionality reduction Techniques

```
graph TD; A[Dimensionality reduction Techniques] --> B[Feature Selection]; A --> C[Dimensionality Reduction]; B --> D["● Missing Value Ratio<br>● Low Variance Filter<br>● High Correlation Filter<br>● Random Forest<br>● Backward Feature Extraction<br>● Forward Feature Selection"]; C --> E[Components/Factors based]; C --> F[Projection Based]; E --> G["● Factor Analysis<br>● Principal Component Analysis<br>● Independent Component Analysis"]; F --> H["● ISOMAP<br>● t-SNE<br>● UMAP"];
```

## Feature Selection

- Missing Value Ratio
- Low Variance Filter
- High Correlation Filter
- Random Forest
- Backward Feature Extraction
- Forward Feature Selection

## Dimensionality Reduction

### Components/Factors based

- Factor Analysis
- Principal Component Analysis
- Independent Component Analysis

### Projection Based

- ISOMAP
- t-SNE
- UMAP

# Methods of Dimensionality Reduction

## Feature Extraction Methods:

- **Principal Component Analysis (PCA):** A linear technique that identifies the directions (principal components) that capture the maximum variance in the data.
  - **Advantages:** Widely used, easy to understand.
  - **Disadvantages:** Can be sensitive to scaling; sometimes, the number of components is unknown.
- **Linear Discriminant Analysis (LDA):** A supervised technique that finds a linear combination of features that separates two or more classes.
- **Generalized Discriminant Analysis (GDA):** A non-linear extension of LDA.

# Feature Selection

- Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy.
- **Feature selection** is a process that chooses a sub set of  $M$  features from the original set of  $N$  features ( $M \leq N$ ) so that the feature space is optimally reduced according to a certain criterion
- The role of feature selection in machine learning is
  - to reduce the dimensionality of feature space
  - to speed up a learning algorithm
  - to improve the predictive accuracy of a classification algorithm and
  - to improve the comprehensibility of the learning results

# Methods of Dimensionality Reduction

## Feature Selection Methods:

- **Filter Methods:** Use statistical measures to score features.
  - **Techniques:** Correlation, Chi-Square Test, ANOVA, Information Gain.
- **Wrapper Methods:** Use a machine learning model to evaluate subsets of features.
  - **Techniques:** Forward Selection, Backward Elimination.
- **Embedded Methods:** The feature selection is part of the model training process.
  - **Techniques:** LASSO, Ridge Regression.

# Feature Selection

## 1. Filters Methods

- Features are selected on the basis of statistics measures
- Does not depend on the learning algorithm and chooses the features as a pre-processing step
- Filters out the irrelevant feature and redundant columns from the model by using different metrics
- Needs low computational time and does not overfit the data.

### **Some common techniques of filters method are:**

- Missing Value Ratio
- Chi-Square Test
- ANOVA
- Information Gain, etc.

# Feature Selection

## 2. Wrappers Methods

- In this method, **some features are fed to the ML model, and evaluate the performance.**
- The performance decides **whether to add those features or remove to increase the accuracy of the model.** This method is more accurate than the filtering method but complex to work.

**Some common techniques of wrapper methods are:**

- **Forward Selection**
- **Backward Selection**
- **Bi-directional Elimination**

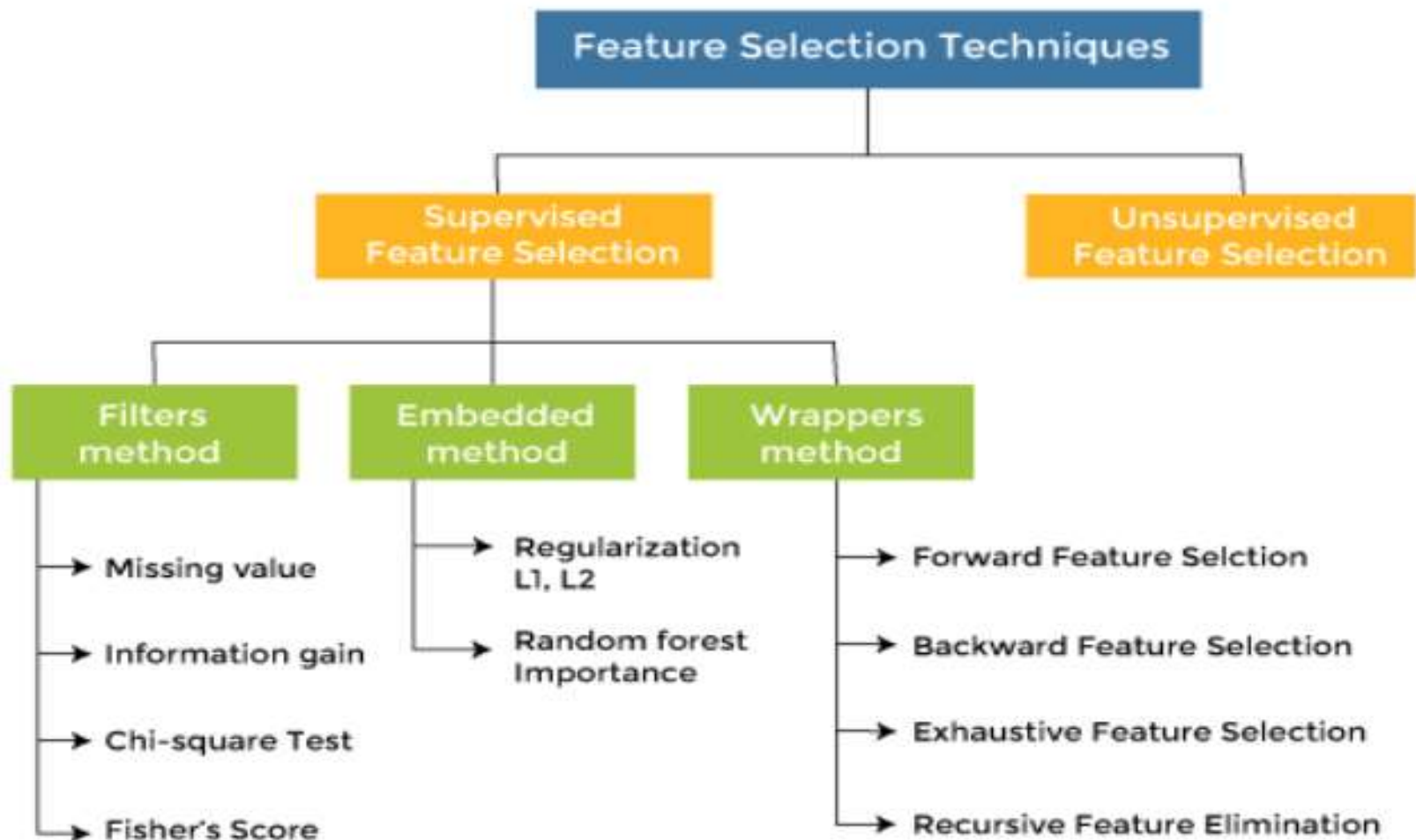
## Feature Selection

### 3. Embedded Methods:

- Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature.
- Embedded methods combined the advantages of both filter and wrapper methods by considering the interaction of features along with low computational cost.
- These are fast processing methods similar to the filter method but more accurate than the filter method.
- These methods are also iterative, which evaluates each iteration, and optimally finds the most important features that contribute the most to training in a particular iteration.

**Some common techniques of Embedded methods are:**

- **Regularization L1,L2.**
- **Random Forest Importance etc.**





# Feature Extraction:

- Feature extraction refers to the process of **transforming raw data into numerical features** that can be processed while preserving the information in the original data set.
- Feature extraction generates new variables by extracting them from the raw data.
- By performing feature extraction, the relevant features are separated (“extracted”) from the irrelevant ones.
- With fewer features to process, the dataset becomes simpler and the accuracy and efficiency of the analysis improves.
- The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling.
- Feature extraction can be accomplished manually or automatically

**Some common feature extraction techniques are:**

- **Principal Component Analysis**
- **Linear Discriminant Analysis**
- **Kernel PCA**
- **Quadratic Discriminant Analysis**

# Feature Extraction:

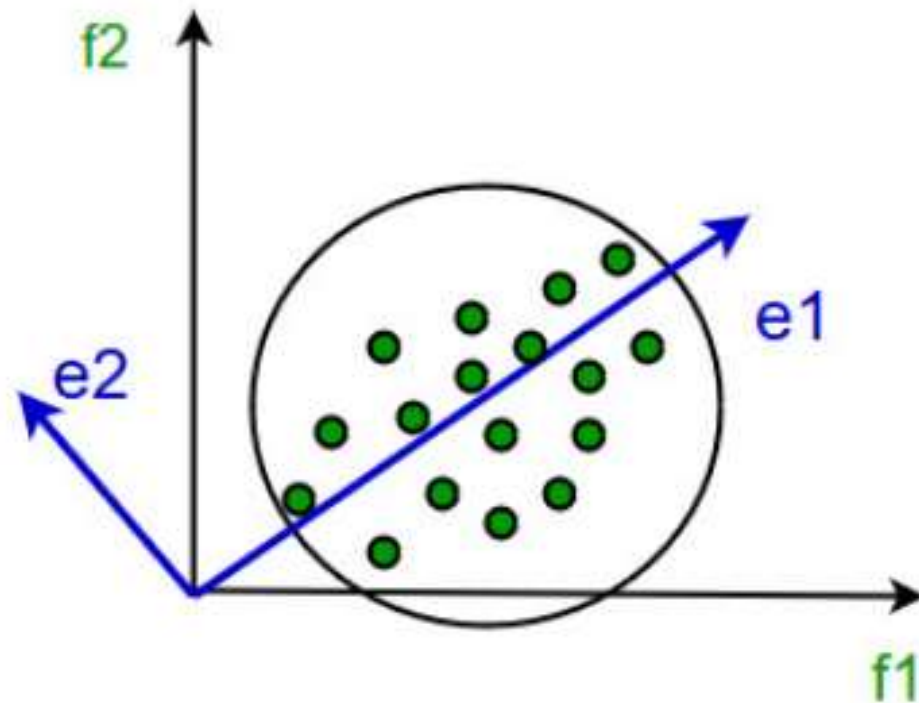
- Feature extraction can be accomplished manually or automatically
- **Manual feature extraction** requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features.
- In many situations, having a good understanding of the background or domain can help make informed decisions as to which features could be useful.
- **Automated feature extraction** uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention.
- This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms.
- For image data, feature extraction has been largely replaced by the first layers of deep networks
- Feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA).

# Common techniques of Dimensionality Reduction

- **Principal Component Analysis**
- **Backward Elimination**
- **Forward Selection**
- **Score comparison**
- **Missing Value Ratio**
- **Low Variance Filter**
- **High Correlation Filter**
- **Random Forest**
- **Factor Analysis**
- **Auto-Encoder**

# Principal Component Analysis

- This method was introduced by Karl Pearson. It works on a condition that **while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.**



# Principal Component Analysis

- Principal Component Analysis is a statistical process that **converts the observations of correlated features into a set of linearly uncorrelated features** with the help of orthogonal transformation. **These new transformed features are called the Principal Components.**
- It is one of the popular tools that is used for **exploratory data analysis** and **predictive modelling**.
- PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality.
- Some real-world applications of PCA are **image processing, movie recommendation system, optimizing the power allocation in various communication channels.**

# Principal Component Analysis

- The PCA algorithm is based on some mathematical concepts such as:
- **Variance and Covariance**
- **Eigenvalues and Eigen factors**

## Principal Component Analysis

- **Variance** refers to the **spread of a data set around its mean value.**
- It is calculated by finding the probability-weighted average of squared deviations from the expected value.
- While performing market research, variance is particularly useful when calculating probabilities of future events.
- Variance is a great way to find all of the possible values and likelihoods that a random variable can take within a given range.

## Principal Component Analysis

- **A variance value of zero represents that all of the values within a data set are identical, while all variances that are not equal to zero will come in the form of positive numbers.**
- **The larger the variance, the more spread in the data set.**
- **A large variance means that the numbers in a set are far from the mean and each other**
- **A small variance means that the numbers are closer together in value.**



## Principal Component Analysis

- **X** represents an individual data point,
- **u** represents the mean of the data points,
- **N** represents the total number of data points.
- Note that while calculating a sample variance in order to estimate a population variance, the denominator of the variance equation becomes  $N - 1$ .
- This removes bias from the estimation, as it prohibits the researcher from underestimating the population variance.

$$\sigma^2 = \frac{\Sigma (x - \mu)^2}{N}$$

# Principal Component Analysis

## An Advantage of Variance

- One of the primary advantages of variance is that it treats all deviations from the mean of the data set in the same way, regardless of direction.
- **A Disadvantage of Variance**
- it gives added weight to numbers that are far from the mean, or outliers.
- Squaring these numbers can at times result in skewed interpretations of the data set as a whole.

## Principal Component Analysis

- **Covariance** refers to the **measure of the directional relationship between two random variable.**
- A **positive covariance** means that the two variables at hand are **positively related**, and they move in the same direction.
- A **negative covariance** means that the variables are **inversely related**, or that they move in opposite directions.

## Principal Component Analysis

- **X** represents the independent variable,
- **Y** represents the dependent variable,
- **N** represents the number of data points in the sample,
- **x-bar** represents the mean of the **X**, and
- **y-bar** represents the mean of the dependent variable **Y**.

$$COV(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

## Principal Component Analysis

- **Covariance** is used to measure variables that have different units of measurement. So covariance does not use one standardized unit of measurement.
- **Correlation**, on the other hand, standardizes the measure of interdependence between two variables and informs researchers as to how closely the two variables move together.

## Concept of Feature Preprocessing of data:

- **Eigenvalue** - the eigenvalue is a **scalar** that is used to transform the eigenvector.
- The basic equation is  $\mathbf{Ax} = \lambda\mathbf{x}$
- The number or scalar value “ $\lambda$ ” is an eigenvalue of A.

### EigenValue Example

In this shear mapping, the blue arrow changes direction, whereas the pink arrow does not. Here, the pink arrow is an eigenvector because it does not change direction. Also, the length of this arrow is not changed; its eigenvalue is 1.



## Principal Component Analysis

- **Eigenvector-** Eigenvector of a square matrix is defined as a non-vector in which when a given matrix is multiplied.
- If **A** be an  **$n \times n$  matrix** and  **$\lambda$**  be the **eigenvalues** associated with it.
- Then, eigenvector **v** can be defined by the following relation:
- **$A\mathbf{v} = \lambda\mathbf{v}$**

# Principal Component Analysis

## Eigenvector Examples

**Example:** Find the eigenvector of the given matrix:  $(A - \lambda I) X = 0$

$$A = \begin{bmatrix} 1 & 4 \\ -4 & -7 \end{bmatrix}$$

**Solution:**

Given:

$$A = \begin{bmatrix} 1 & 4 \\ -4 & -7 \end{bmatrix}$$

$$|A - \lambda I| = \begin{vmatrix} 1 - \lambda & 4 \\ -4 & -7 - \lambda \end{vmatrix}$$

$$(1 - \lambda)(-7 - \lambda) - 4(-4) = 0$$

$$(\lambda + 3)^2 = 0$$

Therefore,  $\lambda = -3, -3$

Use the eigenvector equation

$$AX = \lambda X$$

Substitute  $\lambda$  value in the equation:

$$AX = -3X$$

We know that,

$$\left( \begin{bmatrix} 1 & 4 \\ -4 & -7 \end{bmatrix} + \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$4x + 4y = 0$$

Or

$$x + y = 0$$

Assume that  $x = k$

So, it becomes

$$k + y = 0$$

$$y = -k$$

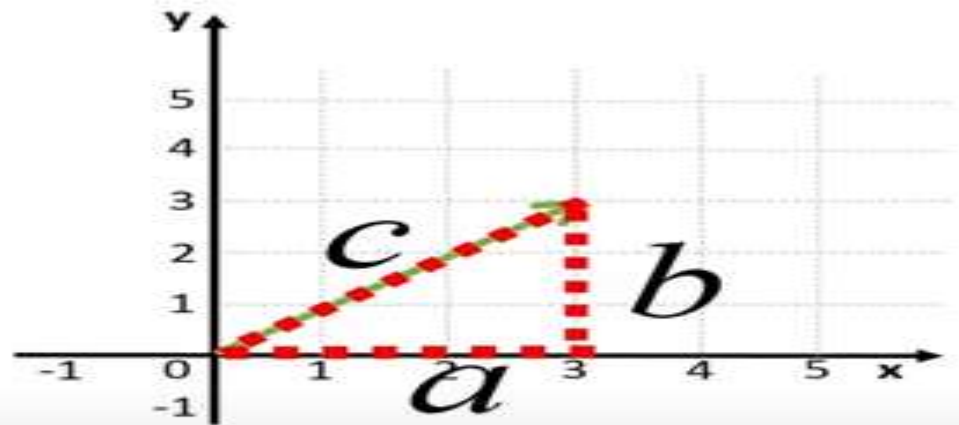
Therefore, the eigenvector is

$$X = \begin{bmatrix} x \\ y \end{bmatrix} = k \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



# Principal Component Analysis

$$v = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

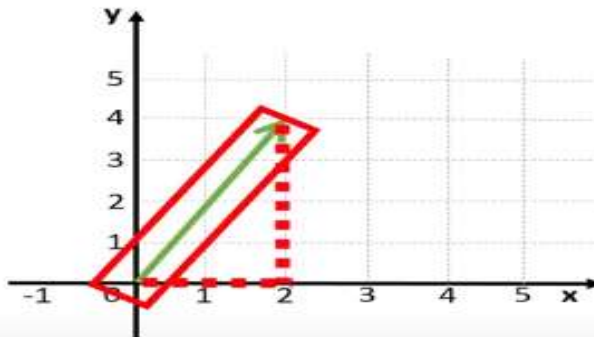


Pythagoras' theorem

$$c^2 = a^2 + b^2$$

Vectors

$$v = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$



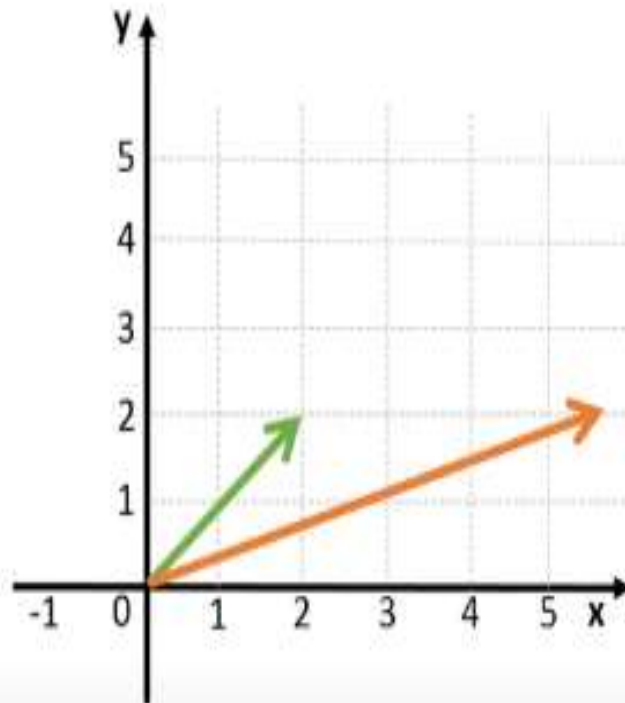
$$c = \sqrt{2^2 + 4^2} = \sqrt{20}$$

The length of this vector is the square root of 20, since the length of the two dashed lines of this triangle are two and four.

# Eigenvectors

$$v = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$$



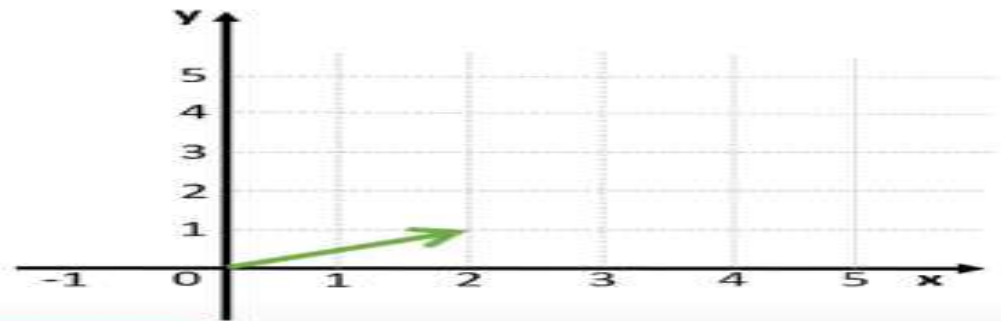
$$Av = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$

Since this vector changes its direction when multiplied with matrix A, we can conclude that this vector is not an eigenvector of matrix A.

# Eigenvectors

$$v = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$$

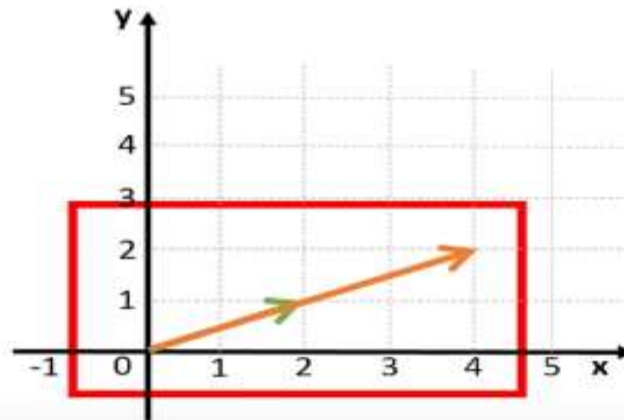


$$Av = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Eigenvalue

$$v = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$$



$$Av = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

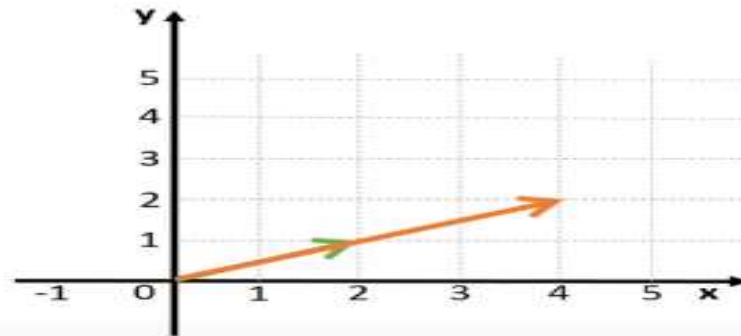
If we draw this vector in the plane, we see that the new vector has the same direction as the original vector. Thus, vector  $v$  is therefore an eigenvector of matrix  $A$ .

## Eigenvalue

$$v = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$$

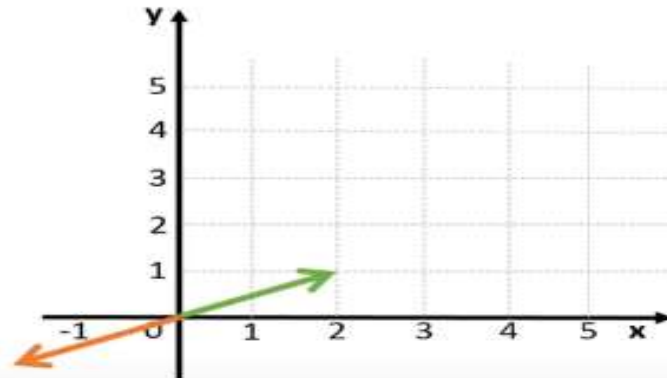
$$Av = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \boxed{2} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$



$$\lambda = 2$$

two is the eigenvalue,

## Eigenvalue



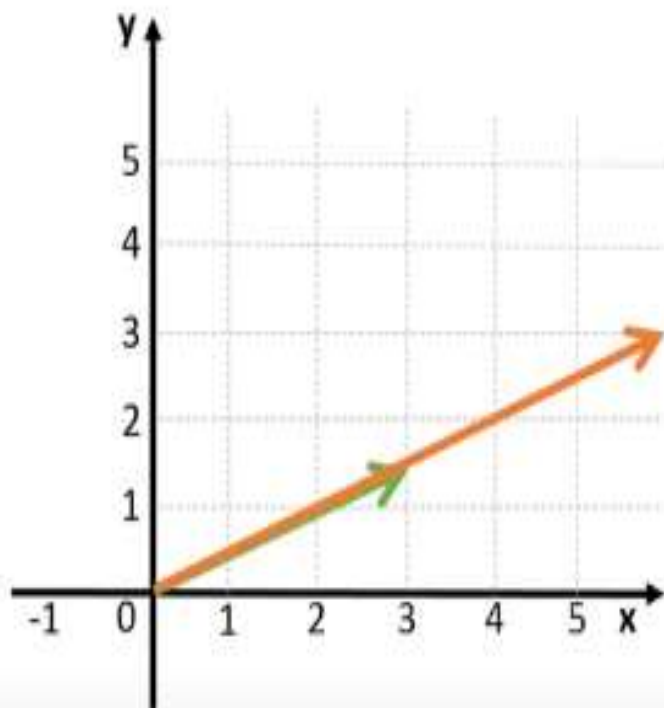
$$Av = \boxed{\lambda} v$$

Note that, the eigenvalue can even be negative, which means that the direction of the new vector is reversed but still on the same "line".

# Eigenvectors

$$v = \begin{bmatrix} 3 \\ 1.5 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$$



$$Av = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix} = 2 \begin{bmatrix} 3 \\ 1.5 \end{bmatrix}$$

Note that, although we have found a new eigenvector, the corresponding eigenvalue does not change. We see that the eigenvalue is still 2.

# Principal Component Analysis

## Advantages of Dimensionality Reduction

- It helps in **data compression**, and hence **reduced storage space**.
- It **reduces computation time**.
- It also helps **remove redundant features**, if any.

## Disadvantages of Dimensionality Reduction

- It may lead to some amount of **data loss**.
- PCA tends to find **linear correlations between variables**, which is sometimes undesirable.
- PCA fails in cases where **mean and covariance** are not enough to define datasets.
- We may not know how many principal components to keep- in practice, some thumb rules are applied.

## Some common terms used in PCA algorithm:

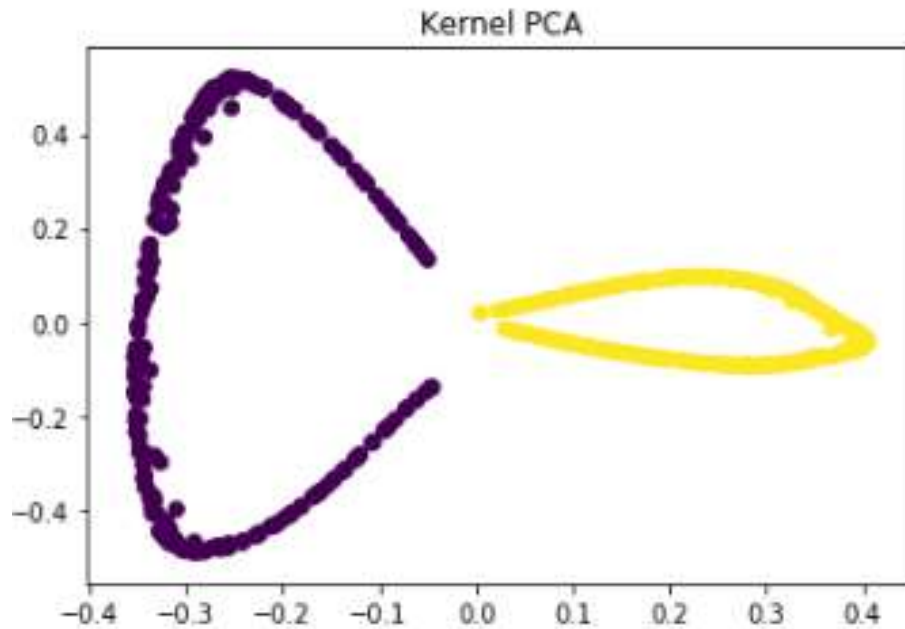
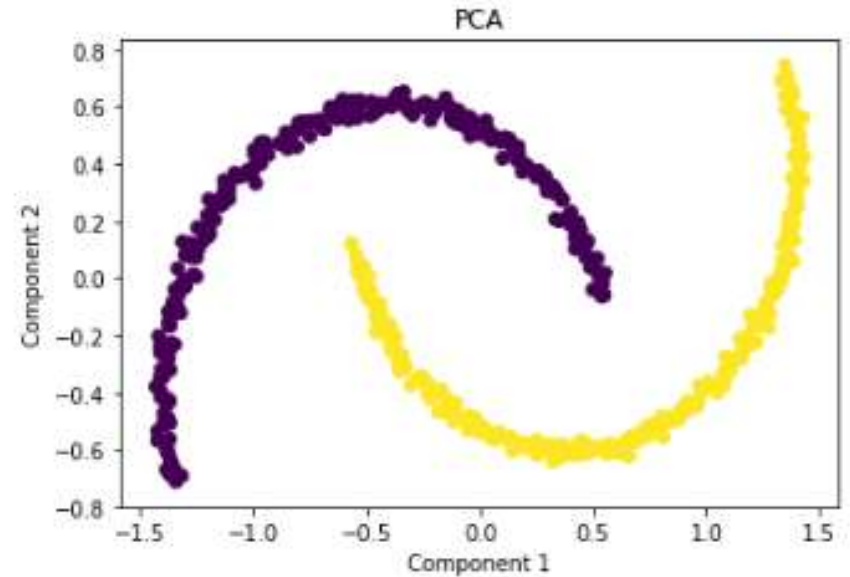
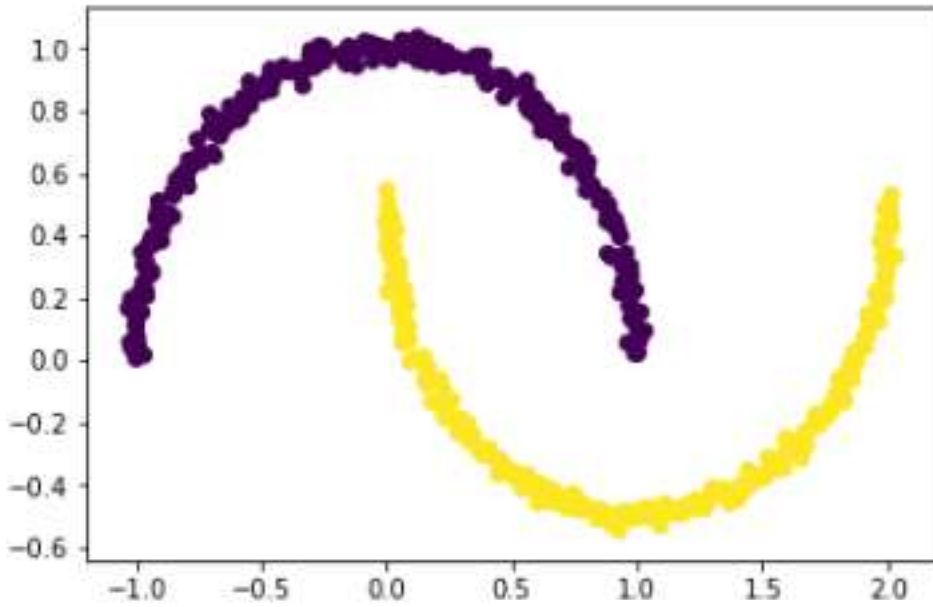
- **Dimensionality:** It is the **number of features** or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that **how strongly two variables are related to each other**. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that **variables are not correlated to each other**, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a **square matrix  $M$** , and a non-zero vector  $v$  is given. Then  $v$  will be eigenvector if  $Av$  is the **scalar multiple of  $v$** .
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

## Kernel PCA

- PCA is a **linear method**. That is it can only be applied to datasets which are **linearly separable**. It does an excellent job for datasets, which are linearly separable.
- But, if we use it to non-linear datasets, we might get a result which may not be the optimal dimensionality reduction.
- Kernel PCA uses a **kernel function to project dataset** into a higher dimensional feature space, where it is linearly separable. It is similar to the idea of Support Vector Machines.
- There are various kernel methods like linear, polynomial, and Gaussian.
- In the kernel space the two classes are linearly separable. **Kernel PCA uses a kernel function to project the dataset into a higher-dimensional space, where it is linearly separable.**



# Kernel PCA



## Local Binary Pattern

- is in its ability to differentiate tiny differences in texture and topography, to identify key features with which we can then differentiate between images of the same type — no painstaking labeling required.
- **The goal of LBP is to encode geometric features of an image by detecting edges, corners, raised or flat areas and hard lines; allowing us to generate a feature vector representation of an image, or group of images.**

## Local Binary Pattern

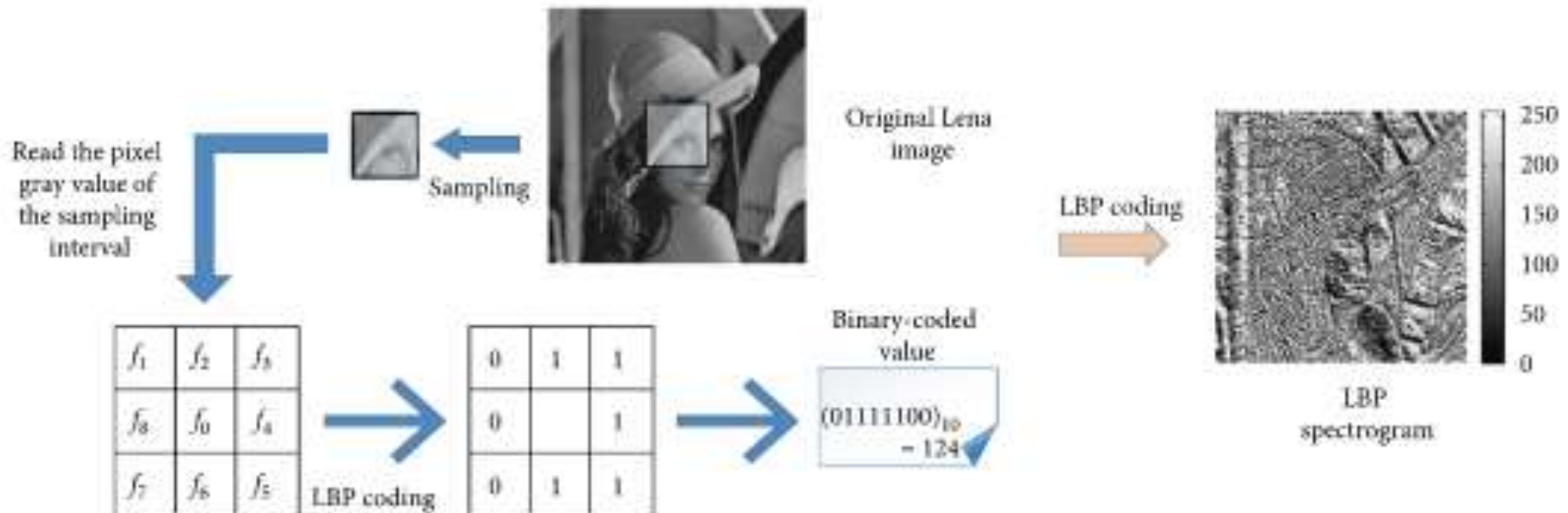
- we can determine the level of similarity between our target representation and an unseen image and can **calculate the probability that the image presented** is of the same variety or type as the target image.
- **LBP can be split into 4 key steps:**
  - · Simplification
  - · Binarisation
  - · PDF (probability density function) calculation
  - · Comparison (of the above functions)
- *Simplification*

## Local Binary Pattern

- This is our data preprocessing step. In essence, this is our first step in dimensionality reduction, which allows our algorithm to focus purely on the local differences in luminance, rather than worrying about any other potential features.
- Therefore, we first convert our image into a single channel (typically greyscale) representation
- **Binarisation**
- Next, we calculate the relative local luminance changes. **This allows us to create a local, low dimensional, binary representation of each pixel based on luminance.**

# Local Binary Pattern

- For each comparison, we output a binary value of 0 or 1, dependent on whether the central pixel's intensity (scalar value) is greater or less (respectively) than the comparison pixel.
- This forms a  $k$ -bit binary value, which can then be converted to a base 10 number; forming a new intensity for that given pixel.



# Selection vs. Extraction

- In **feature selection** we try to find the best subset of the input feature set.
- In **feature extraction** we create new features based on transformation or combination of the original feature set.
- Both selection and extraction lead to the dimensionality reduction.
- No clear cut evidence that one of them is superior to the other on all types of task.

Feature  
Extraction

Color Features

Histogram  
Color Moment  
Correlogram

Shape  
Features

Region Based  
method

Contour Based  
Method

Texture  
Features

Spatial  
Texture

Spectral  
texture



# Why to do it?

1. We're interested in features – we want to know which are relevant. If we fit a model, it should be interpretable.
  - facilitate data visualization and data understanding
  - reduce experimental costs (measurements)
2. We're interested in prediction – features are not interesting in themselves, we just want to build a good predictor.
  - faster training
  - defy the curse of dimensionality



# Classification of FS methods

- Filter
  - Assess the relevance of features only by looking at the intrinsic properties of the data.
  - Usually, calculate the feature relevance score and remove low-scoring features.
- Wrapper
  - Bundle the search for best model with the FS.
  - Generate and evaluate various subsets of features. The evaluation is obtained by training and testing a specific ML model.
- Embedded
  - The search for an optimal subset is built into the classifier construction (e.g. decision trees).

# Filter methods

- Two steps (score-and-filter approach)
  1. assess each feature individually for its potential in discriminating among classes in the data
  2. features falling beyond threshold are eliminated
- Advantages:
  - easily scale to high-dimensional data
  - simple and fast
  - independent of the classification algorithm
- Disadvantages:
  - ignore the interaction with the classifier
  - most techniques are univariate (each feature is considered separately)

# Wrappers

- Search for the best feature subset in combination with a fixed classification method.
- The goodness of a feature subset is determined using cross-validation ( $k$ -fold, LOOCV)
- Advantages:
  - interaction between feature subset and model selection
  - take into account feature dependencies
  - generally more accurate
- Disadvantages:
  - higher risk of overfitting than filter methods
  - very computationally intensive

# Sequential Forward Selection

- SFS
- At the beginning select the best feature using a scalar criterion function.
- Add one feature at a time which along with already selected features maximizes the criterion function.
- A **greedy** algorithm, cannot retract (also called nesting effect).

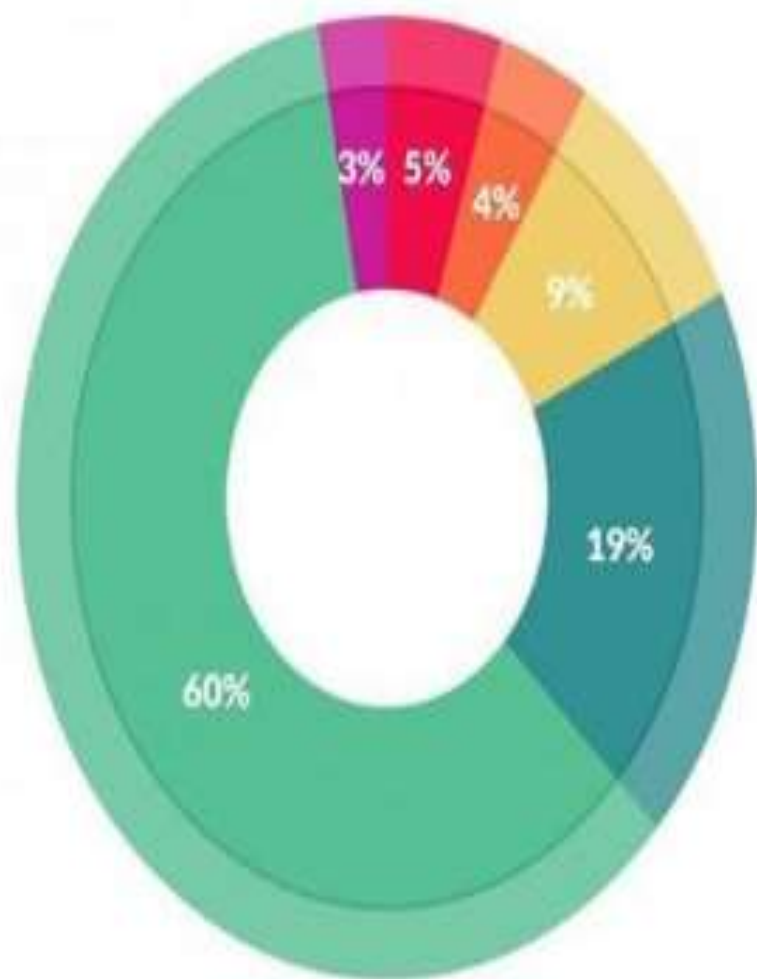
# Sequential Backward Selection

- SBS
- At the beginning select all  $d$  features.
- Delete one feature at a time and select the subset which maximize the criterion function.
- Also a greedy algorithm, cannot retract.
- Complexity is  $O(d)$ .

## Statistical feature engineering:

- **Feature engineering** refers to a process of **selecting & transforming** variables/features in your dataset when creating a **predictive model** using machine learning.
- Therefore you have to extract the features from the **raw dataset** you have collected before training your data in machine learning algorithms.
- Feature engineering has two goals:
- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the **performance** of machine learning models.





What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

CrowdFlower Survey

## Statistical feature engineering:

- **Mean:** The "average" number; found by adding all data points and dividing by the number of data points.
- Example: The mean of 444, 111, and 777 is  $(4+1+7)/3 = 12/3 = 4$   
 $(4+1+7)/3=12/3=4$  left parenthesis, 4, plus, 1, plus, 7, right parenthesis, slash, 3, equals, 12, slash, 3, equals, 4.



## Statistical feature engineering:

- **Median:** The middle number; found by ordering all data points and picking out the one in the middle (or if there are two middle numbers, taking the mean of those two numbers).
- Example: The median of 444, 111, and 777 is 444 because when the numbers are put in order (1(left parenthesis, 1, 444, 7)7)7, right parenthesis, the number 444 is in the middle.

## Statistical feature engineering:

- **Mode:** The most frequent number—that is, the number that occurs the highest number of times.
- Example: The mode of  $\{4, 222, 444, 333, 222, 2\}$  is 222 because it occurs three times, which is more than any other number.

## Feature Vector Creation.

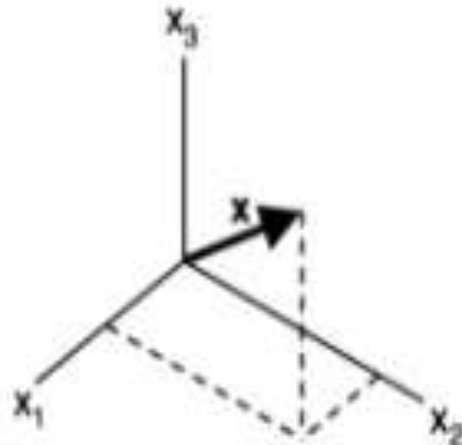
- A vector is a series of numbers, like a matrix with one column but multiple rows, that can often be represented spatially.
- A **feature** is a numerical or symbolic property of an aspect of an object.
- A **feature vector** is a vector containing **multiple elements about an object**. Putting feature vectors for objects together can make up a **feature space**.

## Statistical feature engineering:

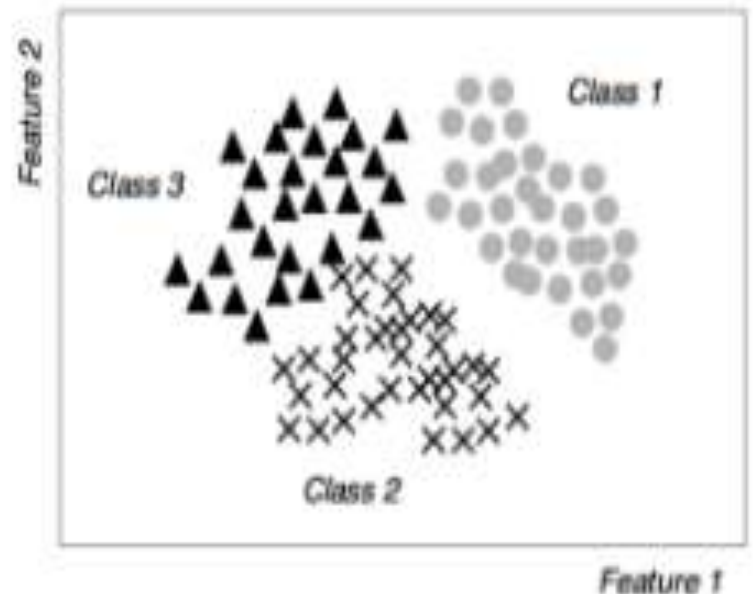
- The features may represent, as a whole, one mere pixel or an entire image. The granularity depends on what someone is trying to learn or represent about the object. You could describe a 3-dimensional shape with a feature vector indicating its height, width, depth, etc.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

**Feature vector**



**Feature space (3D)**



**Scatter plot (2D)**

## Statistical feature engineering:

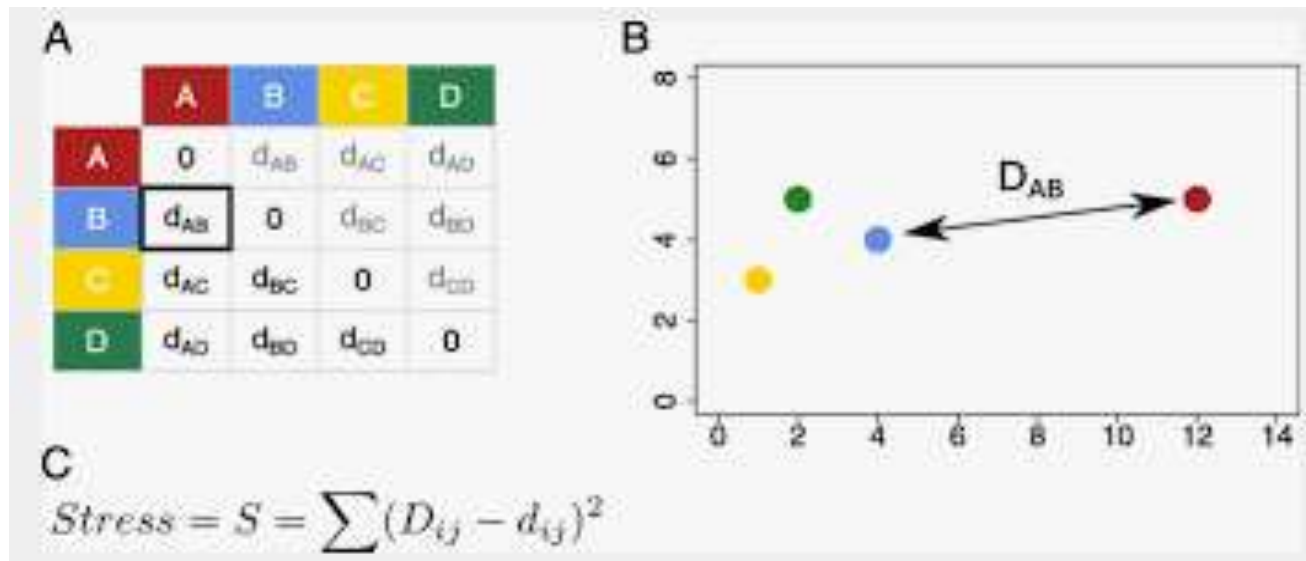
- One simple way to compare the feature vectors of two objects is to take the [Euclidean distance](#).
- In **image processing**, features can be gradient magnitude, color, grayscale intensity, edges, areas, and more.
- Feature vectors are particularly popular for analyses in image processing because of the convenient way attributes about an image, like the examples listed, can be compared numerically once put into feature vectors.
- In **speech recognition**, features can be sound lengths, noise level, noise ratios, and more.

- In **spam-fighting initiatives**, features are abundant. They can be IP location, text structure, frequency of certain words, or certain email headers.
- Feature vectors are used in [classification problems](#), [artificial neural networks](#), and [k-nearest neighbors](#) algorithms in machine learning.
- **Feature vector is an n-dimensional vector of numerical features that describe some object in pattern recognition in machine learning**

## What is Multidimensional Scaling?

- Multidimensional scaling is a **visual representation of distances or dissimilarities between of high dimensional data.**
- The “**multidimensional**” part is due to the fact that you aren’t limited to **two dimensional graphs or data. Three-dimensional, four-dimensional and higher plots are possible.**
- Its use in geostatistics helps visually assess and understand multivariate data in a lower dimension.

- By reducing the dimensionality of the data one can observe patterns, gradients, and clusters that may be helpful in exploratory data analysis.
- MDS does this by projecting the multivariate distances between entities to a best-fit configuration in lower dimensions.





Model listening data as a product of latent factors

