ARTIFICIAL INTELLIGENCE

DATA HANDLING - NOISE REDUCTION

CO -2

Session - 18

AIM

To familiarize the students with the concept of Principal component analysis (PCA)

INSTRUCTIONAL OBJECTIVES

This session is designed to:

1. **Noise reduction , outlier detection , imputation (KNN)**
2. Understand the Nature of Noise in Data
3. Explore the Impact of Noise on Data Analysis
4. Classify Types of Noise

LEARNING OUTCOMES

At the end of this session, students will be able to:

Define and Identify Data Noise

Differentiate Between Types of Noise

Apply Noise Detection Techniques

What is FFT?

- **Fast Fourier Transform (FFT)** is a mathematical algorithm used to convert **time-domain signals** into the **frequency domain**.
- It helps reveal the **frequency components** of a signal, making it easier to identify and remove **high-frequency noise**.

Why Use FFT for Noise Reduction?

- **Noise often resides in high frequencies**, while important data patterns lie in lower frequencies.
- By transforming data into the frequency domain, we can:
  - **Identify noise frequencies**
  - **Filter them out**
  - **Reconstruct a cleaner signal**

Steps for Noise Reduction Using FFT:

- **Input the Time-Domain Data**
  - For example: Sensor readings, ECG signal, audio wave, etc.
- **Apply FFT**
  - Use tools like numpy.fft.fft() in Python.
  - This transforms your signal into a spectrum of frequencies.
- **Analyze the Frequency Spectrum**
  - Plot the magnitude spectrum to observe peaks (signal) and clutter (noise).
- **Apply Frequency Filtering**
  - **Low-pass filter**: Retain low-frequency components (main signal), remove high-frequency noise.
  - Set a **threshold**: Zero out frequencies above a certain cutoff.
- **Apply Inverse FFT (IFFT)**
  - Convert the filtered frequency spectrum back into the time domain using numpy.fft.ifft().
  - Result: **Smoothed, denoised signal**.

Advantages of FFT for Noise Reduction

- Effective for **periodic signals** (audio, biomedical, vibration data)
- Allows **precise frequency filtering**
- Computationally efficient for large datasets

What Is an Outlier?

- An **outlier** is a data point that lies far outside the range of the majority of other values in a dataset.
  These can distort statistical analyses and models, so detecting them is an essential step in **data handling**.

Box Plot Overview (a.k.a. Box-and-Whisker Plot)

- A **box plot** is a graphical representation of data distribution using **five-number summary**:
- **Minimum**
- **First Quartile (Q1)** – 25th percentile
- **Median (Q2)** – 50th percentile
- **Third Quartile (Q3)** – 75th percentile
- **Maximum**
- Outliers are detected using the **IQR (Interquartile Range)**:
- **IQR = Q3 − Q1**
- **Lower Bound** = Q1 − 1.5 × IQR
- **Upper Bound** = Q3 + 1.5 × IQR
- Any data point **outside this range** is considered an **outlier**.

Benefits of Box Plot for Outlier Detection

- Easy to visualize **spread and symmetry**
- Effective for **univariate outlier detection**
- Identifies **mild** and **extreme** outliers

Imputation Using Measures of Centrality

- **Definition:**
- **Central tendency** methods replace missing values with a representative value of the dataset:
- **Mean** – average of all values
- **Median** – middle value (robust to outliers)
- **Mode** – most frequent value (used for categorical data)
- 🔧 **When to Use:**
- **Mean**: Symmetric distributions without outliers
- **Median**: Skewed distributions or presence of outliers
- **Mode**: Categorical or discrete values

Imputation Using KNN (K-Nearest Neighbors)

- KNN imputation replaces missing values by finding the **k most similar (nearest) data points** and imputing the missing values using their feature values (mean, median, or weighted average).
- **When to Use:**
- Mixed data types
- Non-linear patterns
- Assumes data points with similar features likely have similar values

How It Works:

- For each instance with missing values:
  - Compute **Euclidean distance** (or other) to all complete instances
  - Find the **K-nearest neighbors**
  - Replace missing value with the **mean (or median)** of those neighbors' corresponding values