Assignment No. 4

Problem Statement: Implement normal distribution in python and visualize it for Mean =100, standard _deviation =4, dataset size=100000

Objective: To understand and implement the normal distribution in Python, visualize it for a dataset of size 100,000 with a mean of 100 and a standard deviation of 4, and learn the basics of normalization.

Prerequisite :

1. Basic understanding of probability and statistics
2. Familiarity with Python programming
3. Knowledge of libraries like NumPy and Matplotlib for numerical computations and plotting.

Theory :

The normal distribution, also known as the Gaussian distribution, is a continuous probability distribution characterized by its bell-shaped curve. It is symmetric about the mean, with data near the mean more frequent in occurrence than data far from it.

Key properties:

- Mean (μ): The average or central value.
- Standard Deviation (σ): Measures the spread of data points from the mean.
- 68-95-99.7 Rule: Approximately 68% of data falls within 1 standard deviation, 95% within 2, and 99.7% within 3.

Normalization

Normalization is a technique to scale data such that it fits a specific range, typically [0, 1] or [-1, 1]. When applied using the Z-score formula:

Z=(X−μ)σZ = \frac{(X - \mu)}{\sigma}Z=σ(X−μ)

where XXX is the data point, μ\muμ is the mean, and σ\sigmaσ is the standard deviation, it converts data into a standard normal distribution (mean 0, standard deviation 1).

Algorithm (if any to achieve the objective)

1.Generate Random Data: Use the NumPy library to create a dataset of size 100,000 with a mean of 100 and a standard deviation of 4.

2.Visualize the Distribution: Plot the histogram using Matplotlib.

3.Apply Normalization (optional): Convert the data to a standard normal form.

4.Plot Normalized Data: Compare the original and normalized data distributions.

References :

https://medium.com/@snehabajaj108/the-normal-distribution-with-python-2cb3bf57ee47

Conclusion

Normal distribution is crucial for statistical analysis. The implementation above provides a basic understanding of how to generate, visualize, and normalize data in Python, which can be extended for various real-world applications like data preprocessing, hypothesis testing, and predictive modeling.