Aliah University

আলিয়া বিশ্ববিদ্যালয়

جامعة عالية

COMPUTER SCIENCE AND ENGINEERING

ASSIGNMENT

GROUP NO. : 1

MEMBERS : RAJASREE LAHA, RUPESH THAKUR, JISHAN ALAM

ROLL : CSE214002, CSE214007, CSE214013

PROGRAMME : B.TECH

COURSE TITLE : MACHINE LEARNING & SOFT COMPUTING LAB

COURSE CODE : CSEUGPC25

COURSE FACULTY : DR. SAIYED UMER

YEAR : 4TH

SEMESTER : 7TH

Project Title	**Consider a random vector, a collection of random values of n-dimension [min 'n' = 100]. Fit Poisson's distribution and Gaussian distribution for these collected values.**
Student Details	1. CSE214002, Rajasree Laha 2. CSE214007, Rupesh Thakur 3. CSE214013, Jishan Alam
Machine Configuration	GPU: N/A RAM: 8.00 GB (7.65 GB usable) OS: 22631.4317 Processor: 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz 2.50 GHz
Database description	**Database name:** Random Vector Data **Number of samples:** 100 (minimum), here it is 150(dimention of the vector with random values in between 1 & 100) **Number of features for each sample:** 1 (since we are fitting distributions to a collection of random values) **Number of classes:** N/A (this is not a classification problem) **Type:** Regression (since we are fitting continuous distributions like Poisson and Gaussian)
Objectives	In this analysis, we aim to fit both **Poisson** and **Gaussian (Normal)** distributions to a set of randomly generated values. The goal is to compare how well each distribution models the data, visualize the fit, and interpret the results.
Methodology	We will use the following techniques: • **Poisson Distribution:** Fitting using the mean of the data as the Poisson parameter λ. • **Gaussian Distribution:** Fitting by calculating the mean (μ) and

<table>
<tr><td></td><td>standard deviation (σ) of the data.

• Data Visualization: Plotting both the Poisson PMF and Gaussian PDF on top of the data’s histogram to visually assess the fit.</td></tr>
<tr><td>Process</td><td>1. Data Generation: We generate a random vector containing 150 values, where each value is a random integer between 1 and 100.

2. Fitting Distributions:
● Poisson: We compute the mean of the data to set the Poisson parameter λ.

● Gaussian: We calculate both the mean (μ) and standard deviation (σ) of the data to fit the Gaussian distribution.

3. Visualization: A histogram of the data is plotted, and both the Poisson PMF and Gaussian PDF are overlaid on the plot for comparison.

4. Analysis: We visually assess how well each distribution fits the data based on the plotted curves.</td></tr>
<tr><td>Experimental Results</td><td><table>
<tr><th>Distribution</th><th>Parameters</th><th>Fit (Visual Assessment)</th></tr>
<tr><td>Poisson</td><td>$\lambda = 49.95$</td><td>Moderate fit</td></tr>
<tr><td>Gaussian</td><td>$\mu = 49.95, \sigma = 28.09$</td><td>Better fit</td></tr>
</table></td></tr>
<tr><td>Discussion</td><td>The Gaussian distribution shows a better fit to the data compared to the Poisson distribution, which is expected because the Gaussian distribution is typically more suitable for continuous data. The Poisson distribution works better for discrete event counts, which may not fully represent this dataset. The smoother nature of the Gaussian curve reflects the variation in the data more accurately.</td></tr>
</table>

Problem Statement: Consider a random vector, a collection of random values of n-dimension [min ‘n’ = 100]. Fit Poisson's distribution and Gaussian distribution for these collected values.

```
# import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats



# Generate random values for an n-dimensional vector (n >= 100)
n = 150    #n>=100
random_vector = np.random.randint(1, 100, n)   # Generate an n-dimensional array of random integers between 1 and 100
np.random.seed(42)   # For reproducibility



# Fit Poisson distribution
lambda_poisson = np.mean(random_vector)   # Poisson distribution is parameterized by its mean (λ)



# Fit Gaussian (Normal) distribution
mu, std = np.mean(random_vector), np.std(random_vector)   # Gaussian is parameterized by mean and std dev(standard deviation)



# Visualization
# Plot histogram of the random vector
plt.hist(random_vector, bins=20, density=True, alpha=0.6, color='g', label='Data')
```

#creates a histogram of the random_vector data with 20 bins, normalizes the height to form a probability density(Probability density is a way to show how likely it is for a value to occur, with higher values meaning more likely, and the total area under the curve always adding up to 1.),

#adds 60% transparency, sets the color to green, and labels it as 'Data'.

Generate points for plotting fitted Poisson and Gaussian curves

x = np.arange(0, 150)   #creates an array of values from 0 to 149 to use as the x-axis for plotting

poisson_pmf = stats.poisson.pmf(x, lambda_poisson)   # Poisson PMF,calculates the Poisson probability mass function (PMF) at each value in x using lambda_poisson

#The Poisson probability mass function (PMF) gives the probability of a given number of events happening in a fixed interval of time or space,

#assuming the events occur independently and at a constant average rate (λ). It's used for counting events, like the number of phone calls a call center gets in an hour

gaussian_pdf = stats.norm.pdf(x, mu, std)   # Gaussian PDF,calculates the Gaussian (Normal) probability density function (PDF) at each value in x using the mean mu and standard deviation std

#A probability density function (PDF) shows how likely different values of a continuous random variable are,

#with the area under the curve representing the probability of the variable falling within a specific range. The total area under the curve equals 1

Plot Poisson fit

plt.plot(x, poisson_pmf, 'bo', ms=9, label='Poisson Fit (λ={:.2f})'.format(lambda_poisson), alpha=0.6)

#plots the Poisson probability mass function as blue circles ('bo') at the x-values, sets the marker size to 8,

#labels the plot with the calculated λ value, and adds 60% transparency to the points

Plot Gaussian fit

```
plt.plot(x, gaussian_pdf, 'r-', lw=3, label='Gaussian Fit (μ={:.2f}, σ={:.2f})'.format(mu, std), alpha=0.6)

#plots the Gaussian probability density function as a red line ('r-') at the x-values, sets the line width to 2, labels the plot with the calculated mean (μ) and standard deviation (σ),

#and adds 60% transparency to the line



# Add labels and legend

plt.xlabel('Value')

plt.ylabel('Probability Density')

plt.legend(loc='best')   #adds a legend to the plot at the best location, automatically determining the most suitable position to avoid overlapping with the plotted data

plt.title('Fitting Poisson and Gaussian Distributions')



# Show the plot

plt.show()
```