National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Ma'am Neelma Naz **Date:** December 18, 2025

Semester: 7th **Group:** 02

CS471 Machine Learning

Lab 14: Anomaly Detection

		PLO4	PLO5	PLO5	PLO8	PLO9
		CLO4	CLO5	CLO5	CLO6	CLO7
Student Name	**Reg. No**	**Viva / Quiz / Demo** 5 Marks	**Analysis of Data in Report** 5 Marks	**Modern Tool Usage** 5 Marks	**Ethics** 5 Marks	**Individual and Teamwork** 5 Marks
Hanzla Sajjad	403214					
Irfa Farooq	412564					

Introduction

This laboratory exercise will focus on the concept on the technique of anomaly detection which is an unsupervised learning approach used to detect outliers in a dataset.

Objectives

The following are the main objectives of this lab:

- Implement a Gaussian distribution function
- Perform anomaly detection from scratch to detect outliers

Lab Conduct

- Respect faculty and peers through speech and actions
- The lab faculty will be available to assist the students. In case some aspect of the lab experiment is not understood, the students are advised to seek help from the faculty.
- In the tasks, there are commented lines such as #YOUR CODE STARTS HERE# where you have to provide the code. You must put the code/screenshot/plot between the #START and #END parts of these commented lines. Do NOT remove the commented lines.
- Use the tab key to provide the indentation in python.
- When you provide the code in the report, keep the font size at 12

Theory

Anomaly detection is an unsupervised learning approach used to detect outliers in a dataset. Anomaly detection can be used as preprocessing task for removing unwanted outliers from the dataset and is also useful for fraud detection as fraud activity differs from user activity and can serve as a potential outlier.

A brief summary of the relevant keywords and functions in python is provided below:

print()	output text on console
input()	get input from user on console
range()	create a sequence of numbers
len()	gives the number of characters in a string
if	contains code that executes depending on a logical condition
else	connects with **if** and **elif**, executes when conditions are not met
elif	equivalent to **else if**
while	loops code as long as a condition is true
for	loops code through a sequence of items in an iterable object
break	exit loop immediately
continue	jump to the next iteration of the loop
def	used to define a function
pd.read_csv	import csv file as a dataframe
df.to_csv	export dataframe as a csv file

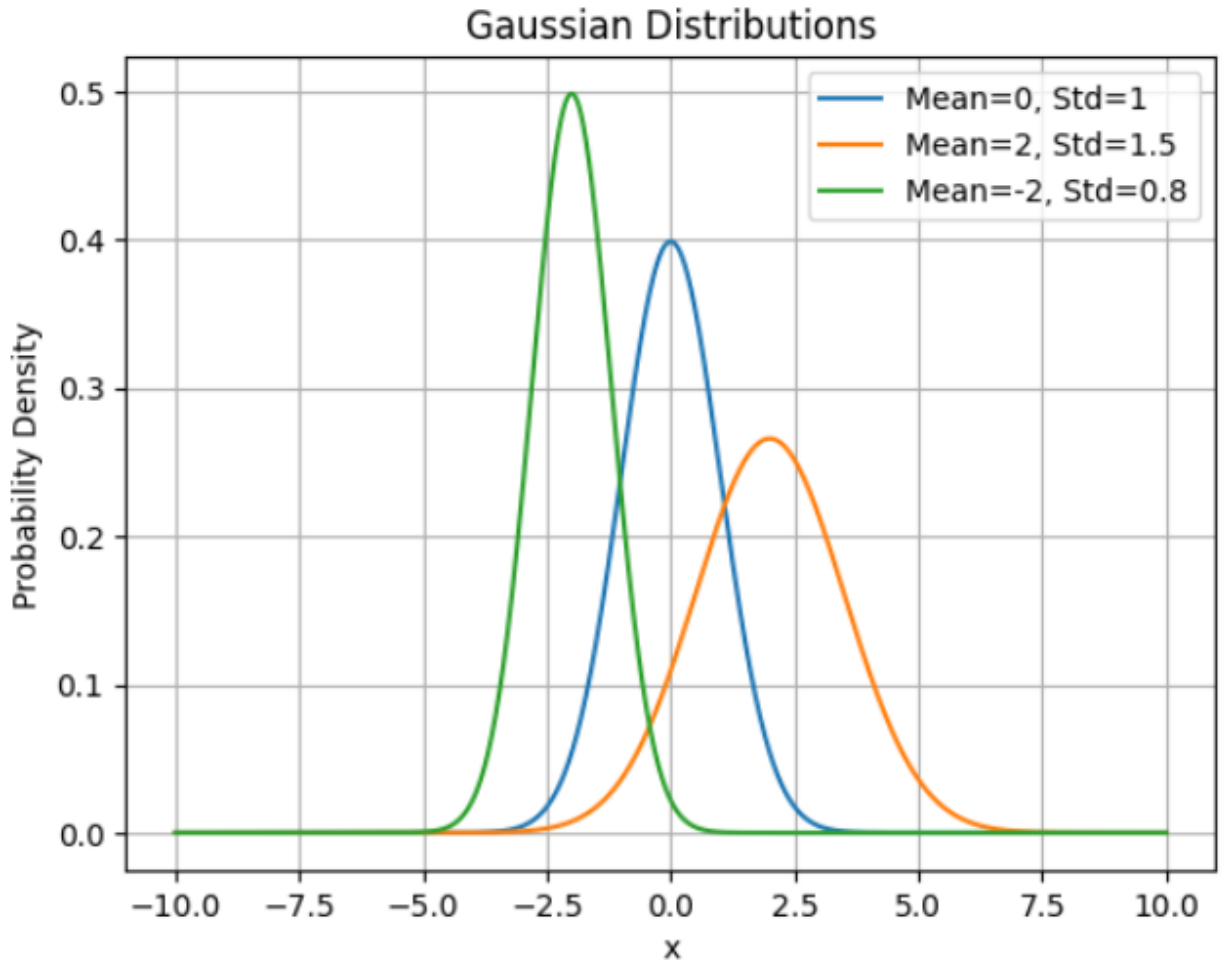
Lab Task 1 – Gaussian Distribution _____

Write a function in python that generates a Gaussian distribution given a mean value and standard deviation value. Provide the codes and the plot at least 3 distributions.

Code

```python
# Importing Important Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Task 1: Gaussian Distribution
def gaussian_distribution(x, mean, std):
    coefficient = 1 / (std * np.sqrt(2 * np.pi))
    exponent = np.exp(-((x - mean) ** 2) / (2 * std ** 2))
    return coefficient * exponent

# Using random x values for plotting
x = np.linspace(-10, 10, 1000)

# Generating 3 distributions
y1 = gaussian_distribution(x, mean=0, std=1)
y2 = gaussian_distribution(x, mean=2, std=1.5)
y3 = gaussian_distribution(x, mean=-2, std=0.8)

# Plotting the data
plt.plot(x, y1, label="Mean=0, Std=1")
plt.plot(x, y2, label="Mean=2, Std=1.5")
plt.plot(x, y3, label="Mean=-2, Std=0.8")
plt.title("Gaussian Distributions")
plt.xlabel("x")
plt.ylabel("Probability Density")
plt.legend()
plt.grid(True)
plt.show()
```

Output Console



Lab Task 2 – Dataset with Outliers _____

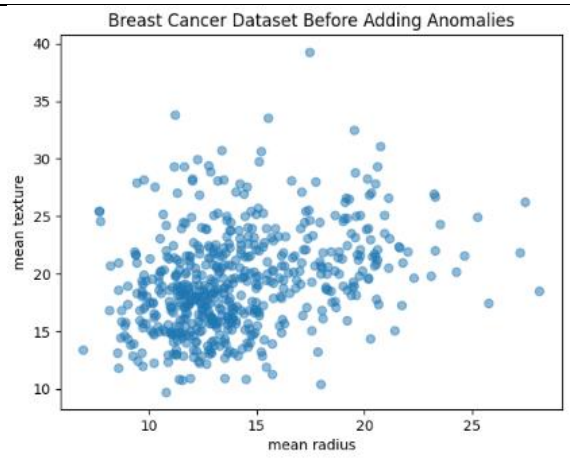
Download a dataset containing at least 3 features and 500 examples. Create a scatter plot of the dataset. Calculate the mean and standard deviations of each feature. Next, add a few anomalies into the dataset and plot it again. Provide the code and all relevant screenshots.

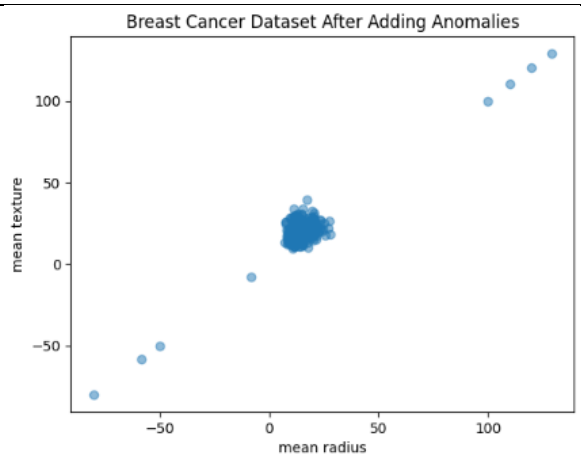
Code

```
# Task 2: Dataset with Outliers
data = load_breast_cancer()
X = data.data

# Convert to DataFrame
df = pd.DataFrame(X, columns=data.feature_names)

# Confirm dataset size
print("Number of samples:", df.shape[0])
print("Number of features:", df.shape[1])

# Calculate mean and standard deviation
print("\nMeans:\n", df.mean())
print("\nStandard Deviations:\n", df.std())

# Scatter plot before adding anomalies
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], alpha=0.5)
plt.title("Breast Cancer Dataset Before Adding Anomalies")
plt.xlabel(df.columns[0])
plt.ylabel(df.columns[1])
plt.show()

# Add anomalies
anomalies = np.array([
    [100] * df.shape[1],
    [-50] * df.shape[1],
    [120] * df.shape[1],
    [-80] * df.shape[1],
    [110] * df.shape[1],
```

```
    [-58] * df.shape[1],
    [129] * df.shape[1],
    [-8] * df.shape[1]
])

df_anomaly = pd.DataFrame(anomalies, columns=df.columns)
df = pd.concat([df, df_anomaly], ignore_index=True)

# Scatter plot after adding anomalies
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], alpha=0.5)
plt.title("Breast Cancer Dataset After Adding Anomalies")
plt.xlabel(df.columns[0])
plt.ylabel(df.columns[1])
plt.show()
```

Output Console





Lab Task 3 – Anomaly Detection _____

Write a program in python from scratch to detect anomalies from a dataset. Use the Gaussian distribution function to calculate the probability of each example. Using the calculated probabilities, make a scatter plot in which the outliers are highlighted.

Code

```
# Task 3: Anomaly Detection
# Use dataset AFTER adding anomalies
X = df.values

# Mean and std of each feature
mean = np.mean(X, axis=0)
std = np.std(X, axis=0)

# Compute probability for each example
probabilities = np.ones(X.shape[0])
for i in range(X.shape[1]):
    probabilities *= gaussian_distribution(X[:, i], mean[i], std[i])

# Set threshold dynamically as a small percentile of probabilities
epsilon = np.percentile(probabilities, 1.5)  # bottom 1.5% as anomalies
outliers = probabilities < epsilon

# Scatter plot highlighting anomalies
plt.scatter(X[~outliers, 0], X[~outliers, 1], alpha=0.5, label="Normal")
plt.scatter(X[outliers, 0], X[outliers, 1], color="red", label="Anomalies")

plt.title("Anomaly Detection using Gaussian Distribution")
plt.xlabel(df.columns[0])
plt.ylabel(df.columns[1])
plt.legend()
plt.show()
```

Output Console