

USECASE-2

Develop an application to perform plotting probability using matplotlib.

An application to perform probability plotting using Matplotlib can be developed by defining the probability distribution (either analytically or empirically) and then visualizing it.

Here is an example demonstrating how to plot a normal distribution's Probability Density Function (PDF) and Cumulative Distribution Function (CDF) using Matplotlib and NumPy:

Program:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from scipy.stats import norm
```

```
# 1. Define the distribution parameters (for a normal distribution)
```

```
mu = 0 # Mean
```

```
sigma = 1 # Standard deviation
```

```
# 2. Generate x-values for plotting
```

```
x = np.linspace(-4, 4, 1000) # 1000 points between -4 and 4
```

```
# 3. Calculate the Probability Density Function (PDF)
```

```
pdf_values = norm.pdf(x, loc=mu, scale=sigma)
```

```
# 4. Calculate the Cumulative Distribution Function (CDF)
```

```
cdf_values = norm.cdf(x, loc=mu, scale=sigma)
```

```
# 5. Create the plots
```

```
plt.figure(figsize=(10, 5))
```

```
# Plotting the PDF
```

```
plt.subplot(1, 2, 1) # 1 row, 2 columns, first plot
```

```
plt.plot(x, pdf_values, color='blue', label='PDF')
```

```
plt.title('Probability Density Function (PDF)')
```

```
plt.xlabel('X-value')
```

```
plt.ylabel('Probability Density')
```

```
plt.grid(True)
```

```
plt.legend()
```

```
# Plotting the CDF
```

```
plt.subplot(1, 2, 2) # 1 row, 2 columns, second plot
```

```
plt.plot(x, cdf_values, color='red', label='CDF')
```

```
plt.title('Cumulative Distribution Function (CDF)')
```

```
plt.xlabel('X-value')
```

```
plt.ylabel('Cumulative Probability')
```

```
plt.grid(True)
```

```
plt.legend()
```

```
plt.tight_layout() # Adjusts subplot parameters for a tight layout
```

```
plt.show()
```

Short Algorithm: Plot PDF and CDF of Normal Distribution

1. Import Libraries

Use `numpy`, `matplotlib.pyplot`, and `scipy.stats.norm`.

2. Set Parameters

Define `mu = 0` and `sigma = 1`.

3. Generate Data

Create 1000 `x` values from -4 to 4.

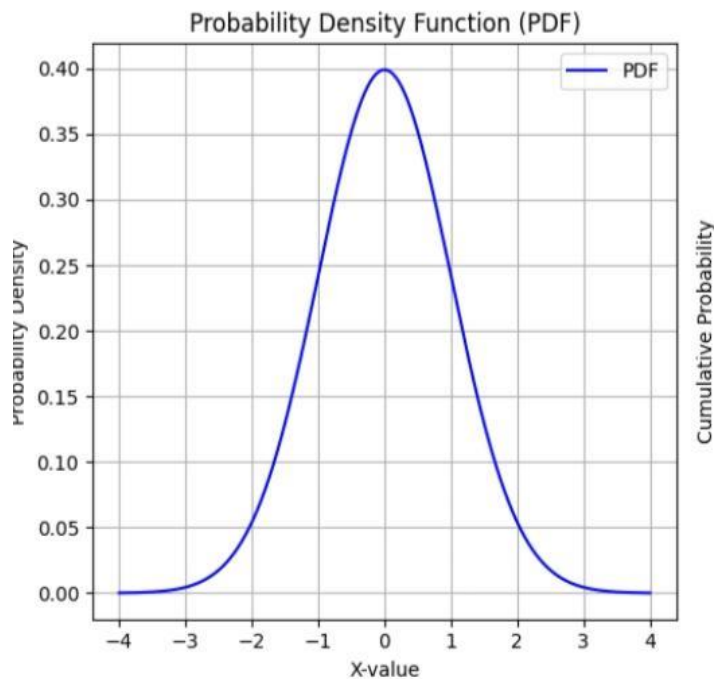
4. Compute Distributions

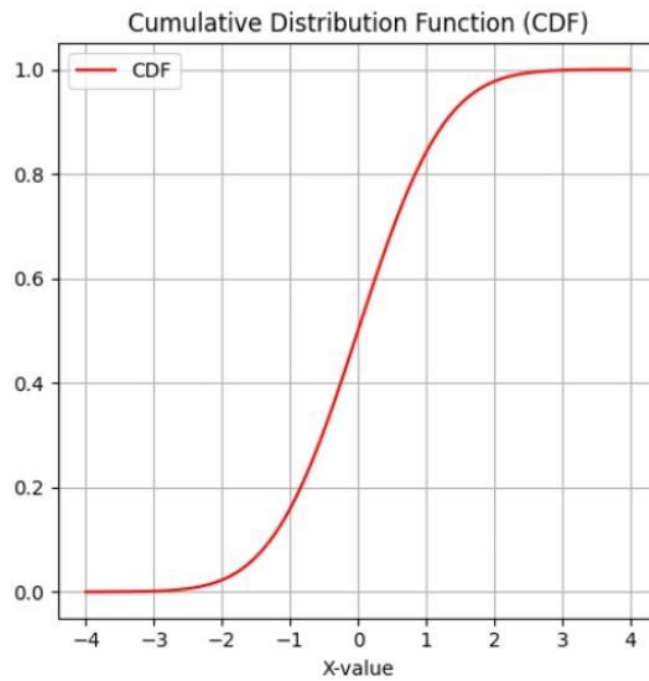
- Calculate **PDF** using `norm.pdf(x, mu, sigma)`.
- Calculate **CDF** using `norm.cdf(x, mu, sigma)`.

5. Plot Results

- Plot **PDF** in the first subplot with labels and grid.
- Plot **CDF** in the second subplot with labels and grid.
- Use `tight_layout()` and `show()` to display both plots.

Output





Result :

Thus to perform the application of plotting probabilities using matplotlib is successfully completed.