

Task 13 : use case**PYTHON PROGRAMMING:****Use Case 2:**

**AIM:** To develop a Python application that plots probability distributions using matplotlib.

**ALGORITHM:**

STEP1: Start the program.

STEP2: Import required libraries:

- numpy for numerical computations.
- matplotlib.pyplot for plotting.
- scipy.stats for probability distributions.

STEP3: Define the range of random variables (x-values).

STEP4: Compute the probability values (y-values) using a chosen probability distribution (e.g., Normal).

STEP5: Plot the probability distribution curve using matplotlib.

STEP6: Add labels, title, legend, and grid for clarity.

STEP7: Display the plot.

STEP8: End the

## PROGRAM:

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import norm, binom, uniform

x = np.linspace(-5, 5, 1000)

mean = 0

std_dev = 1

y_norm = norm.pdf(x, mean, std_dev)

n = 10

p = 0.5

x_binom = np.arange(0, n + 1)

y_binom = binom.pmf(x_binom, n, p)

x_uniform = np.linspace(0, 10, 1000)

y_uniform = uniform.pdf(x_uniform, loc=0, scale=10)

plt.figure(figsize=(10, 6))

plt.subplot(3, 1, 1)

plt.plot(x, y_norm, color='blue', label='Normal Distribution')

plt.title('Probability Distributions using Matplotlib')

plt.ylabel('Probability Density')

plt.legend()

plt.grid(True)

plt.subplot(3, 1, 2)

plt.stem(x_binom, y_binom, basefmt=" ", use_line_collection=True)

plt.ylabel('Probability')

plt.legend(['Binomial Distribution'])
```

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

```
plt.subplot(3, 1, 3)
```

```
plt.plot(x_uniform, y_uniform, color='green', label='Uniform Distribution')
```

```
plt.xlabel('Random Variable (X)')
```

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

```
plt.legend()
```

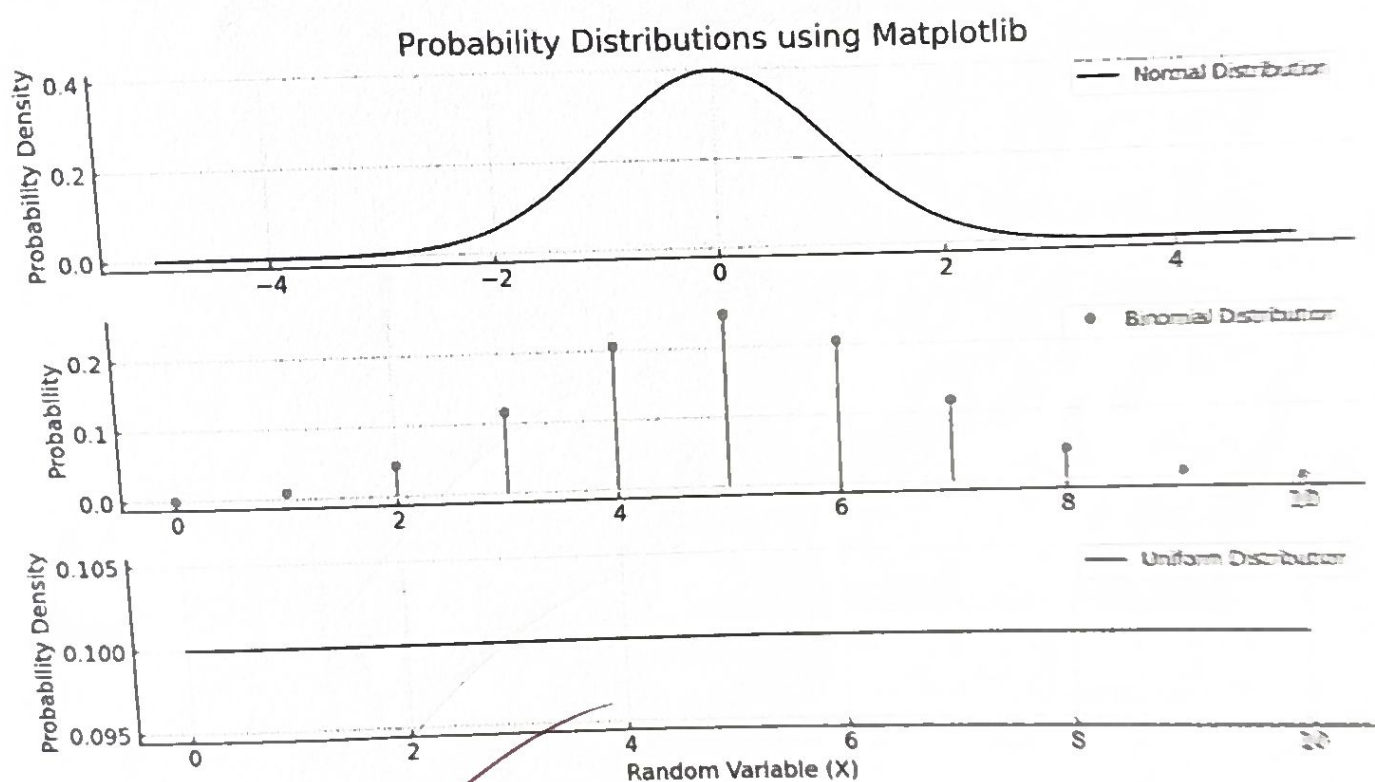
```
plt.grid(True)
```

Display the plot

```
plt.tight_layout()
```

```
plt.show()
```

## OUTPUT IMAGE:



**RESULT:** Thus, the a Python application that plots probability distributions using matplotlib has been executed successfully