

USECASE

Use case: Develop an application to perform Plotting Probabilities Using Matplotlib.

Aim:

To develop a Python application that plots probability distributions (such as Normal, Binomial, or Uniform) using **Matplotlib**, to visualize how probabilities vary across different outcomes.

Procedure:

1. **Import required libraries:**
Import numpy for numerical operations, matplotlib.pyplot for plotting, and scipy.stats for probability functions.
2. **Define the distribution parameters:**
For example, for a **Normal Distribution**, define mean (μ) and standard deviation (σ).
3. **Generate random variables:**
Use NumPy or SciPy to generate a range of x-values that represent possible outcomes.
4. **Compute probability density function (PDF):**
Calculate the corresponding probability values for each x using SciPy's probability distribution functions.
5. **Plot the probability curve:**
Use Matplotlib to plot x (outcomes) vs. probability values (PDF or PMF).
6. **Customize the plot:**
Add title, labels, legend, and grid for clarity.
7. **Display the plot:**
Use plt.show() to visualize the probability distribution.

Program:

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import norm, binom

mean = 0

std_dev = 1

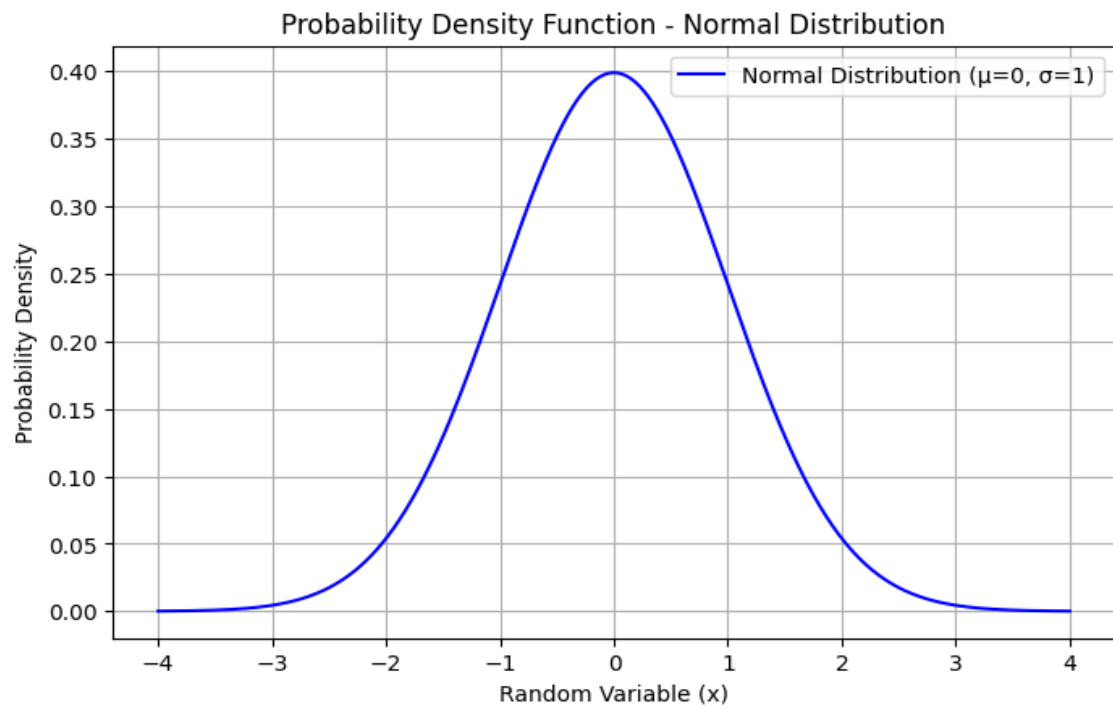
x = np.linspace(-4, 4, 1000)

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

```
plt.figure(figsize=(8, 5))
plt.plot(x, pdf, color='blue', label='Normal Distribution ( $\mu=0$ ,  $\sigma=1$ )')
plt.title("Probability Density Function - Normal Distribution")
plt.xlabel("Random Variable (x)")
plt.ylabel("Probability Density")
plt.legend()
plt.grid(True)
plt.show()

# Parameters
n = 10 # number of trials
p = 0.5 # probability of success
x_binom = np.arange(0, n+1)
pmf = binom.pmf(x_binom, n, p)
plt.figure(figsize=(8, 5))
plt.stem(x_binom, pmf, basefmt=" ", use_line_collection=True)
plt.title("Probability Mass Function - Binomial Distribution (n=10, p=0.5)")
plt.xlabel("Number of Successes")
plt.ylabel("Probability")
plt.grid(True)
plt.show()
```

OUTPUT:



Result:

The application successfully plots probability distributions (Normal and Binomial) using **Matplotlib**, demonstrating how probabilities vary for continuous and discrete random variables.