Department of Computer Science and Engineering
(IoT Cyber Security with Block Chain)
B.Tech. Sem: IV Subject: Statistics for Engineering
Experiment 3

Name: Ashish Maurya **Roll No: B011**

SAP ID: 60019220030 **Batch: B1**

Date:	**07-02-2024**
Aim	3. To Calculate Probability using probability distribution function.
Software	Google Colab
Theory:	**Probability Distribution:** A probability distribution is a mathematical function that describes the likelihood of obtaining the possible outcomes of a random variable. It assigns probabilities to each possible outcome, indicating the likelihood of that outcome occurring. **Types of Probability Distributions:** **1. Discrete Probability Distribution:** A discrete probability distribution lists each possible outcome of a random variable along with its probability of occurrence. The sum of all probabilities in a discrete distribution is equal to 1. Examples: Binomial Distribution, Poisson Distribution. **2. Continuous Probability Distribution:** A continuous probability distribution describes the probabilities of the outcomes of a continuous random variable. Instead of individual values, it assigns probabilities to intervals. Examples: Normal Distribution, Exponential Distribution. Examples with Formulas:

Theory:

3. Binomial Distribution:
Formula: P(X = k) = (n choose k) * p^k * (1-p)^(n-k)
Where:
P(X = k) is the probability of getting exactly k successes.
n is the number of trials.
p is the probability of success on each trial.
(n choose k) is the binomial coefficient.

4. Poisson Distribution:
Formula: P(X = k) = (λ^k * e^(-λ)) / k!
Where:
P(X = k) is the probability of observing k events.
λ (lambda) is the average rate of occurrence of the event.
e is the base of the natural logarithm.
k is the number of occurrences.

5. Cumulative Probability Distribution:
A cumulative probability distribution function gives the probability that the random variable is less than or equal to a certain value. It is obtained by summing the probabilities of all outcomes up to and including a certain value.

6. Normal Distribution:
The normal distribution, also known as the Gaussian distribution, is a continuous probability distribution that is symmetric and bell-shaped. It is characterized by its mean (μ) and standard deviation (σ). The probability density function of the normal distribution is given by the formula:

$$f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where:
μ is the mean of the distribution.
σ is the standard deviation.
e is the base of the natural logarithm.

7. Prior Probability:
Prior probability refers to the initial probability assigned to an event or outcome before considering any additional evidence or information.

8. Posterior Probability:
Posterior probability, in Bayesian statistics, refers to the updated probability of an event occurring after taking into account new evidence or information. It is calculated using Bayes' theorem.

Implementation

Copy problem statement then code and then output.

1. Out of 800 families with 4 children each, how many families would you expect to have (i) 2 boys and 2 girls (ii) at least 1 boy (iii) no girl and (iv) at most 2 girls, assuming equal probability for boys and girls?

```
from scipy.stats import binom
N=800
p1=binom.pmf(k=2,n=4,p=0.5)
print("(i) Total no of families having 2 dogs and 2 cats: ",N*p1
)
p2=binom.pmf(k=0,n=4,p=0.5)
print("(ii) Total no of families having atleast 1 dog: ",N*(1-
p2))
p3=binom.pmf(k=0,n=4,p=0.5)+binom.pmf(k=1,n=4,p=0.5)+binom.pmf(k
=2,n=4,p=0.5)
print("(iii) Total no of families having atmost 2 cats: ",N*p3)
p4=binom.pmf(k=1,n=4,p=0.5)+binom.pmf(k=2,n=4,p=0.5)+binom.pmf(k
=3,n=4,p=0.5)
print("(iv) Total no of families having both cat and dog: ",N*p4
)
```

```
(i) Total no of families having 2 dogs and 2 cats:  300.0
(ii) Total no of families having atleast 1 dog:  750.0
(iii) Total no of families having atmost 2 cats:  550.0
(iv) Total no of families having both cat and dog:  700.0
```

2. The number of yearly breakdowns of a computer is a random variable having Poisson distribution with parameter m=1.8. Find the probability that this computer will function for a year: i) without breakdown. ii) with at most one breakdown

```
from scipy.stats import poisson
m = 1.8
prob_no_breakdown = poisson.pmf(0, m)
print("Probability of functioning without breakdown:", prob_no_breakdown)
prob_at_most_one_breakdown = poisson.cdf(1, m)
print("Probability of functioning with at most one breakdown:", prob_at_most_one_breakdown)
```

```
Probability of functioning without breakdown: 0.16529888822158653
Probability of functioning with at most one breakdown: 0.46283688702044234
```

3. A taxicab company has 12 Ambassadors and 8 Fiats. If 5 of these taxi cabs are in the workshop for repairs and an ambassador is as likely to be in for repairs as a Fiat, what is the probability that

(a) 3 of them are Ambassadors and 2 are Fiats,

(b) at least 3 of them are Ambassadors, and

(c) all the 5 are of the same make?

```
from scipy.stats import binom
total_ambassadors = 12
total_fiats = 8
num_repairs = 5
p_ambassador = total_ambassadors / (total_ambassadors + total_fiats)
p_fiat = total_fiats / (total_ambassadors + total_fiats)
prob_a = binom.pmf(3, num_repairs, p_ambassador) * binom.pmf(2, num_repairs, p_fiat)
print("Probability of 3 Ambassadors and 2 Fiats:", prob_a)
prob_b = 1 - binom.cdf(2, num_repairs, p_ambassador)
print("Probability of at least 3 Ambassadors:", prob_b)
prob_c = binom.pmf(5, num_repairs, p_ambassador) + binom.pmf(5, num_repairs, p_fiat)
print("Probability that all 5 are of the same make:", prob_c)
```

```
Probability of 3 Ambassadors and 2 Fiats: 0.11943935999999998
Probability of at least 3 Ambassadors: 0.68256
Probability that all 5 are of the same make: 0.08799999999999998
```

4. Buses arrive at a specified stop at 15 min intervals starting at 7 a.m., that is, they arrive at 7, 7:15, 7:30, 7:45 and so on. If a passenger arrives at the stop at a random time that is uniformly distributed between 7 a.m. and 7:30 a.m., find the probability that he waits

(i) less than 5 min for a bus and *(ii) at least 12 min for a bus.*

```python
from scipy.stats import uniform
p1 = uniform.cdf(5,0,15)
print("(i) Less than 5 minutes: ",p1)
p2 = uniform.cdf(3,0,15)
print("(ii) Atleast 12 minutes: ",p2)
```

```
(i) Less than 5 minutes:  0.3333333333333333
(ii) Atleast 12 minutes:  0.2
```

5. The mileage which car owners get with a certain kind of radial tire is a RV having an exponential distribution with mean 40,000 km. Find the probabilities that one of these tires will last

(i) at least 20,000 km and *(ii) at most 30,000 km.*

```python
from scipy.stats import expon
import math
mean=40000
prob_at_least_20000= 1-expon.cdf(20000, scale=mean)
print("(i) Probability that the tire will last atleast 20000km: ", prob_at_least_20000)
prob_at_most_30000= expon.cdf(30000,scale=mean)
print("(ii) Probability that the tire will last atmost 30000km: ", prob_at_most_30000)
```

```
(i) Probability that the tire will last atleast 20000km:  0.6065306597126334
(ii) Probability that the tire will last atmost 30000km:  0.5276334472589853
```

6. In a certain city, the daily consumption of electric power in millions of kWh can be treated as a RV having an Erlang distribution with parameter $\lambda = 1/2$ and $k = 3$. If the power plant of this city has a daily capacity of 12 millions kWh, what is the probability that this power supply will be inadequat on any given day.

```python
from scipy.stats import erlang
import math
prob_no_breakdown = poisson.pmf(0, m)
print("Probability of functioning without breakdown:", prob_no_breakdown)
prob_at_most_one_breakdown = poisson.cdf(1, m)
print("Probability of functioning with at most one breakdown:", prob_at_most_one_breakdown)
```

```
Probability of functioning without breakdown: 0.16529888822158653
Probability of functioning with at most one breakdown: 0.46283688702044234
```

7. Suppose, if in a basket there are balls which are defective with a Beta distribution of $\alpha = 2$ and $\beta = 5$. Compute the probability of defective balls in the basket from 20% to 30%.

<table>
<tr>
<td></td>
<td>

```
from scipy.stats import beta
a=2
b=5
lower_bound=0.20
upper_bound=0.30
lower_prob=beta.cdf(lower_bound,a,b)
upper_prob=beta.cdf(upper_bound,a,b)
print("Probability of defective balls from 20% to 30%:", upper_prob - lower_prob)
```

```
Probability of defective balls from 20% to 30%: 0.23518500000000003
```

8. *The marks obtained by a number of students in a certain subject are approximately normally distributed with mean 65 and standard deviation 5. If 3 students are selected at random from this group, what is the probability that at least 1 of them would have scored above 75 ?*

```
from scipy.stats import norm
mean=65
standard_deviation=5
z_score=(75-mean)/standard_deviation
prob_below_75=norm.cdf(z_score)
prob_above_75=1-prob_below_75
prob_atleast_one_above_75=1-(prob_below_75**3)
print("The probability that atleast one of the three students scores above 75 is: ", prob_atleast_one_above_75)
```

```
The probability that atleast one of the three students scores above 75 is:  0.0667094650853094
```

</td>
</tr>
<tr>
<td>Conclusion</td>
<td>Thus, I have successfully implemented and calculated Probability using probability distribution function.</td>
</tr>
</table>

Signature of Faculty