# Hometask 15

Solve **all** parts. Show every important step (formulas, substitution and reasoning) final numeric answers alone will not earn full credit.

**Q1.** A computer–vision model correctly classifies an individual image with probability $p = 0.80$, independently of other images. The number of correctly classified images $X$ follows a Binomial distribution.

(a) Ten images are evaluated. Write the probability mass function (pmf) of $X$.

(b) Calculate $P(X \geq 8)$.

(c) Find $E[X]$ and $\mathrm{Var}(X)$.

(d) If at least 9 correct classifications are required for a pass, what is the probability the batch passes?

**Q2.** In a data-science experiment, a team retrains a lightweight model until it achieves validation accuracy of at least 95%. Each training run has probability $p = 0.30$ of success. The number of runs $Y$ needed to achieve the second success follows a Negative Binomial distribution.

(a) Define the probability mass function (pmf) of $Y$.

(b) Compute $P(Y = 5)$.

(c) Find $E[Y]$.

**Q3.** A labelled dataset contains $N = 500$ text samples, out of which $K = 120$ are labelled as *spam*. You randomly select $n = 25$ samples without replacement. The number of spam samples $Z$ in the selection follows a Hypergeometric distribution.

(a) Write the probability mass function (pmf) of $Z$.

(b) Find $P(Z = 6)$.

(c) Compute $E[Z]$. How would this expectation differ if sampling were done with replacement?

**Q4.** Incoming support tickets to a help-desk arrive following a Poisson process with an average rate $\lambda = 4$ tickets per hour. Let $N$ be the number of tickets arriving.

(a) Find the probability that no tickets arrive in the next 30 minutes.

(b) Calculate the probability that more than 6 tickets arrive in one hour.

(c) Find the mean and variance of the number of tickets arriving in a two-hour window.

**Q5.** Write a Python script that simulates the following process: A population consists of 20 objects, with 7 successes. You draw 5 objects randomly without replacement. Simulate this process 1000 times and plot the distribution of the number of successes. (Use `numpy.random.hypergeometric`.)