**Practical No: 3**

**Implement Bayes Theorem using Python**

**AIM:** Suppose we are given the probability of Mike has a cold as 0.25, the probability of Mike was observed sneezing when he had cold in the past was 0.9 and the probability of Mike was observed sneezing when he did not have cold as 0.20. Find the probability of Mike having a cold given that he sneezes.

**Code:**

def bayes\_theorem(p\_h, p\_e\_given\_h, p\_e\_given\_not\_h):

p\_not\_h = 1 - p\_h

p\_e = (p\_e\_given\_h \* p\_h) + (p\_e\_given\_not\_h \* p\_not\_h)

p\_h\_given\_e = (p\_e\_given\_h \* p\_h) / p\_e

return p\_h\_given\_e

p\_h = float(input("Enter the probability of NK having a cold: "))

p\_e\_given\_h = float(input("Enter the probability of observing sneezing when NK has a cold: "))

p\_e\_given\_not\_h = float(input("Enter the probability of observing sneezing when NK does not have a cold: "))

result = bayes\_theorem(p\_h, p\_e\_given\_h, p\_e\_given\_not\_h)

print("NK's probability of having a cold given that he sneezes (P(H|E)) is:", round(result, 2))

OUTPUT

A screenshot of a computer

Description automatically generated

**b) AIM:** Suppose that a test for using a particular drug is 97% sensitive and 95% specific. That is, the test will produce 97% true positive results for drug users and 95% true negative results for non-drug users. These are the pieces of data that any screening test will have from their history of tests. Bayes’ rule allows us to use this kind of data-driven knowledge to calculate the final probability. Suppose we also know that 0.5% of the general population are users of the drug. What is the probability that a randomly selected individual with a positive test is a drug user?

**Code:**

def drug\_user(prob\_th=0.5,sensitivity=0.97,specificity=0.95,prevelance=0.005,verbose=True):

#FORMULA

p\_user = prevelance

p\_non\_user = 1-prevelance

p\_pos\_user = sensitivity

p\_neg\_user = specificity

p\_pos\_non\_user = 1-specificity

num = p\_pos\_user\*p\_user

den = p\_pos\_user\*p\_user+p\_pos\_non\_user\*p\_non\_user

prob = num/den

print("Probability of the NK being a drug user is", round(prob,3))

if verbose:

if prob > prob\_th:

print("The NK could be an user")

else:

print("The NK may not be an user")

return prob

drug\_user()

**OUTPUT:**

A white background with black text

Description automatically generated