AIDI 1000 - 01 Assignment

In [1]:
```
import pandas as pd
import numpy as np
from scipy.stats import norm
import math
from math import comb
```

Q1: Recent crime statistics collected over the past year reveal the following figures. 81% of all people arrested were male, 15% of all people arrested were under the age of 18, 8% of all people arrested were females and under the age of 18.

Let M represent the event that the person arrested is male, and U18 represents the event that the person arrested is under the age of 18.

Part (a): Complete the table below. Use two decimals in each of your answers. (9 points)

	M	M^c	Row Probabilities
U18			
$U18^c$			
Column Probabilities			

Part (b): A person who was arrested in the past year is randomly chosen. What is the probability that this person is male or under the age of 18? Use two decimals in your answer. (2 points)

Part (c): Find the probability that the person chosen in Part (b) is neither male not under the age of 18. Use two decimals in your answer. (1 points)

Part (d): What is the probability that all people arrested in the past year are male and at least 18 years of age? Use two decimals in your answer. (1 points)

Part (e): Are the events M and U18 mutually exclusive? Select the most appropriate reason below. (2 points) A: M and U18 are not mutually exclusive events, because P(M ∩ U18) ≠ P(M)*P(U18) B: M and U18 are mutually exclusive events, because P(M ∪ U18) = 0 C: M and U18 are mutually exclusive events, because P(M ∩ U18) = P(M)*P(U18) D: M and U18 are mutually exclusive events, because P(M ∩ U18) = 0 E: M and U18 are not mutually exclusive events, because P(M ∩ U18) ≠ 0

Answers :

Part (a) :

Given:

1. P(M) = 0.81
2. P(U18) = 0.15
3. P(Female ∩ U18) = 0.08

Consider M = total percentage of males and Mc = total percentage of female, U18 = total percentage of people arrested are under 18 and U18c = total percentage of people arrested are above.

From above given we can find the missing values to complete the table:

1. Male under 18,
2. male above 18,
3. female above 18

In [2]:
```
total_male = 0.81  # Total percentage of males arrested
total_female = 1 - total_male # Total percentage of female arrested

total_under_18 = 0.15  # Total percentage of people under 18 arrested
total_above_18 = 1 - total_under_18  # Total percentage of people above 18 arrested

female_under_18 = 0.08  # Percentage of females under 18 arrested
```

In [3]:
```
# Calculating male under 18 using total under 18 and female under 18 percentages
male_under_18 = total_under_18 - female_under_18
```

In [4]:
```
# Calculating other probabilities
male_above_18 = total_male - male_under_18
female_above_18 = total_female - female_under_18
```

In [5]:
```
print("Probabilities table : ", '\n')

print(" ".ljust(25), "M".ljust(10), "Mc".ljust(10), "Row Probabilities".ljust(10))
```

```
print("U18".ljust(24), str(round(male_under_18, 4)).ljust(10), str(female_under_18).ljust(15), str(total_under_18).lju

print("U18c".ljust(24), str(round(male_above_18, 4)).ljust(10), str(round(female_above_18, 4)).ljust(15), str(total_ab

print("column Probabilities".ljust(24), str(total_male).ljust(10), str(round(total_female, 4)).ljust(15), str(1.00).lj
```

```
Probabilities table :

                          M          Mc         Row Probabilities
U18                      0.07       0.08            0.15
U18c                     0.74       0.11            0.85
column Probabilities     0.81       0.19            1.0
```

Part B : Probability that the person is male or under the age of 18

Here :

P(M) = total_male P(U18) = total_under_18 P(M ∩ U18) = male_under_18

In [6]:
```
# P(M ∪ U18) = P(M) + P(U18) - P(M ∩ U18)

person_is_male_or_U18 = (total_male + total_under_18 - male_under_18)*100
print(f"Probability that a person is male or under the age of 18: {person_is_male_or_U18:.2f} % ")
```

```
Probability that a person is male or under the age of 18: 89.00 %
```

Part C : Probability that the person is neither male nor under the age of 18

In [7]:
```
# Probability that a person is neither male nor under the age of 18, i.e., female and at least 18 years old.

# Using complement rule: P(A') = 1 - P(A)

person_neither_male_nor_U18 = (1 - person_is_male_or_U18)*100

print(f"Probability that a person is neither male nor under the age of 18: {person_neither_male_nor_U18:.2f} % ")
```

```
Probability that a person is neither male nor under the age of 18: -8800.00 %
```

Part D : Probability that all people arrested in the past year are male and at least 18 years old

In [8]:
```
# Probability that all people arrested in the past year are males and at least 18 years old.

arrested_all_males_at_least_18 = (male_above_18)*100

print(f"Probability that all people arrested are male and at least 18 years old: {arrested_all_males_at_least_18:.2f}
```

```
Probability that all people arrested are male and at least 18 years old: 74.00 %
```

Part E : Part (e): Are the events M and U18 mutually exclusive? Select the most appropriate reason below. (2 points)

A: M and U18 are not mutually exclusive events, because P(M ∩ U18) ≠ P(M)*P(U18)

B: M and U18 are mutually exclusive events, because P(M ∪ U18) = 0

C: M and U18 are mutually exclusive events, because P(M ∩ U18) = P(M)*P(U18)

D: M and U18 are mutually exclusive events, because P(M ∩ U18) = 0

E: M and U18 are not mutually exclusive events, because P(M ∩ U18) ≠ 0

The events M and U18 are not mutually exclusive because there are people who are both male and under 18 (the intersection of M and U18 is not empty).

Thus, the correct answer is option : A: M and U18 are not mutually exclusive events, because P(M ∩ U18) ≠ P(M)*P(U18)

Q2: Suppose that 40% of the voters in a city are in favor of a ban of smoking in public buildings. Suppose 5 voters are to be randomly sampled. Find the probability that:

(a) 2 favor the ban. (2 points)

(b) Less than 4 favor the ban. (4 points)

(c) At least 1 favor the ban. (4 points)

Answers :

Part A : Probability that 2 voters favor the ban

In [9]:
```
# Given data
favor_in_ban = 0.40  # Probability of favoring the ban
voters = 5     # Number of voters sampled

a = 2
probability_a = (comb(voters, a) * (favor_in_ban ** a) * ((1 - favor_in_ban) ** (voters - a)))*100
print("Probability that 2 voters favor the ban:", probability_a, '%')
```

```
Probability that 2 voters favor the ban: 34.56 %
```

Part B : Probability that less than 4 voters favor the ban

In [10]:
```
# (b) Probability that less than 4 voters favor the ban
probability_b = 0
for b in range(4):
    probability_b += comb(voters, b) * (favor_in_ban ** b) * ((1 - favor_in_ban) ** (voters - b))
percent_probability_b = (probability_b)*100
print("Probability that less than 4 voters favor the ban:", percent_probability_b, "%")
```

```
Probability that less than 4 voters favor the ban: 91.296 %
```

Part C : Probability that at least 1 voter favors the ban

In [11]:
```
probability_c = (1 - (comb(voters, 0) * (favor_in_ban ** 0) * ((1 - favor_in_ban) ** voters)))*100

print("Probability that at least 1 voter favors the ban:", probability_c, '%')
```

```
Probability that at least 1 voter favors the ban: 92.224 %
```

Q3: A recent survey of employed Canadians found that 40%, or 4-in-10, would find it difficult to meet their financial obligations if their paycheque was delayed by one-week. You are to randomly select two employed Canadians. Compute the probability that:

(a) Both would find it difficult to meet their financial obligations if their paycheque was delayed by oneweek. (2 points)

(b) Neither would find it difficult to meet their financial obligations if their paycheque was delayed by one-week. (2 points)

(c) At least one of the two would find it difficult to meet their financial obligations if their paycheque was delayed by one-week. (2 points)

(d) Suppose you are to randomly pick n-employed Canadians in such a way that the probability of at least one of them will not be able to meet their financial obligations if their paycheque is delayed by one -week is 0.95. Compute the minimum number of employed Canadians you would have to randomly select. In other words, compute the sample size n. (4 points)

Answers :

In [12]:
```
# Probability of an employed Canadian finding it difficult to meet financial obligations
probability_difficult = 0.4

# Probability of an employed Canadian not finding it difficult
probability_not_difficult = 1 - probability_difficult
```

Part A : Probability that both would find it difficult

In [13]:
```
probability_both_difficult = (probability_difficult * probability_difficult)*100
print("Probability that both would find it difficult to meet their financial obligations if their paycheque was delaye
```

```
Probability that both would find it difficult to meet their financial obligations if their paycheque was delayed by o
neweek: 16.0 %
```

Part B : Probability that neither would find it difficult

In [14]:
```
probability_neither_difficult = (probability_not_difficult * probability_not_difficult)*100
print("Probability that neither would find it difficult to meet their financial obligations if their paycheque was de
```

```
Probability that neither would find it difficult to meet their financial obligations if their paycheque was delayed b
y oneweek: 36.0 %
```

Part C : Probability that at least one would find it difficult

In [15]:
```
probability_at_least_one_difficult = (100 - probability_neither_difficult)
print("Probability that at least one would find it difficult to meet their financial obligations if their paycheque wa
```

```
Probability that at least one would find it difficult to meet their financial obligations if their paycheque was dela
yed by one-week: 64.0 %
```

Part D : Minimum number of employed Canadians you would have to randomly select

Probability that at least one would find it difficult should be 0.95 or greater

(1 - probability_not_difficult)^number >= 0.95

Solving for number : number >= log(0.05) / log(0.6)

In [16]:
```
number = math.ceil(math.log(0.05) / math.log(probability_not_difficult))
print("Minimum number of employed Canadians would have to randomly select:", number)
```

```
Minimum number of employed Canadians would have to randomly select: 6
```

Q4: Most graduate schools of business require applicants for admission to take the SAT examination. Scores on the SAT are roughly normally distributed with a mean of 530 and a standard deviation of 110. What is the probability of an individual scoring above 500 on the SAT?

In [17]:
```
# Mean and standard deviation of SAT scores
mean = 530
standard_deviation = 110

# Score for which we want to find the probability
score = 500

# Calculate the z-score
z_score = (score - mean) / standard_deviation


probability_scoring_above_500 = (1 - norm.cdf(z_score))*100

print("Probability of scoring above 500 on the SAT:", round(probability_scoring_above_500, 2), '%')
```

```
Probability of scoring above 500 on the SAT: 60.75 %
```

Q5: The Edwards's Theater chain has studied its movie customers to determine how much money they spend on concessions. The study revealed that the spending distribution is approximately normally distributed with a mean of 4.11 dollar and a standard deviation of 1.37 dollar. What percentage of customers will spend less than 3.00 dollar on concessions?

In [18]:
```
# Mean and standard deviation of spending on concessions
mean_5 = 4.11
standard_deviation_5 = 1.37

# Spending amount for which we want to find the probability
spending = 3.00

# Calculate the z-score
z_score = (spending - mean_5) / standard_deviation_5

# Calculate the probability of spending less than 3.00 dollars
probability_less_than_3 = norm.cdf(z_score)

# Convert probability to percentage
percentage_less_than_3 = probability_less_than_3 * 100

print("Percentage of customers spending less than $3.00 on concessions:", round(percentage_less_than_3, 2), '%')
```

```
Percentage of customers spending less than $3.00 on concessions: 20.89 %
```

Q6: There are three types of coins which have different probabilities of landing heads when tossed:

– Type A coins are fair, with probability 0.5 of heads – Type B coins are bent and have probability 0.6 of heads – Type C coins are bent and have probability 0.9 of heads

Suppose I have a drawer containing 5 coins: 2 of type A, 2 of type B, and 1 of type C.

I reach into the drawer and pick a coin at random. Without showing you the coin, I flip it once and get heads (i.e. event D).

What is the probability it is:

type A (i.e. P(H = A|D))?

Type B (i.e. P(H = B|D))?

Type C (i.e. P(H =C|D))?

Fill out table below with your answers.

hypothesis	prior	likelihood	posterior
H	$P(H)$	$P(D\|H)$	$P(H\|D)$
A			
B			
C			

Answers :

In [19]:
```
# Define the probabilities
probability_of_A = 0.5
probability_of_B= 0.6
probability_of_C = 0.9

A = 2 / 5
B = 2 / 5
C = 1 / 5

# Compute the total probability of observing heads
total_probability = probability_of_A * A + probability_of_B * B + probability_of_C * C

# Compute the probabilities of having each type of coin given heads
probability_of_head_A = ((probability_of_A * A) / total_probability)*100
probability_of_head_B = ((probability_of_B * B) / total_probability)*100
probability_of_head_C = ((probability_of_C * C) / total_probability)*100

# Print the results
print("Probability of type A given heads:", round(probability_of_head_A, 2), '%')
print("Probability of type B given heads:", round(probability_of_head_B, 2), '%')
print("Probability of type C given heads:", round(probability_of_head_C, 2), '%')
```

```
Probability of type A given heads: 32.26 %
Probability of type B given heads: 38.71 %
Probability of type C given heads: 29.03 %
```

In [20]:
```
print("Hypothesis ".ljust(20), "Prior".ljust(10), "Likelihood".ljust(15), "Posterior".ljust(10))
print("-"*60)
print("A".ljust(21), str(round(A, 4)).ljust(13), str(probability_of_A).ljust(12), str(round(probability_of_head_A, 2))

print("B".ljust(21), str(round(B, 4)).ljust(13), str(probability_of_B).ljust(12), str(round(probability_of_head_B, 2))

print("C".ljust(21), str(round(C, 4)).ljust(13), str(probability_of_C).ljust(12), str(round(probability_of_head_C, 2))
```

```
Hypothesis           Prior      Likelihood      Posterior
------------------------------------------------------------
A                     0.4           0.5          32.26
B                     0.4           0.6          38.71
C                     0.2           0.9          29.03
```

Q7: Consider the following dataset of four rows and three features (Malicious, Viagara, Meet) with class labels (ham and spam).

Suppose we see a message having these features M5 = (Malicious = 'yes',Viagara ='no',Meet ='yes'), What is the probability that it is a spam or ham?

S.No	Malicious	Viagara	Meet	class
M_1	yes	yes	yes	spam
M_2	no	no	yes	ham
M_3	yes	no	yes	spam
M_4	no	yes	no	ham

In [24]:
```
data = [
    {'Malicious': 'yes', 'Viagara': 'yes', 'Meet': 'yes', 'Class': 'spam'},
    {'Malicious': 'no', 'Viagara': 'no', 'Meet': 'yes', 'Class': 'ham'},
    {'Malicious': 'yes', 'Viagara': 'no', 'Meet': 'yes', 'Class': 'spam'},
    {'Malicious': 'no', 'Viagara': 'yes', 'Meet': 'no', 'Class': 'ham'}
]

# Create a Pandas DataFrame
messages_df = pd.DataFrame(data)

# Set index with message id (M1, M2, M3, M4)
messages_df.index = ['M1', 'M2', 'M3', 'M4']

messages_df
```

Out[24]:

	Malicious	Viagara	Meet	Class
M1	yes	yes	yes	spam
M2	no	no	yes	ham
M3	yes	no	yes	spam
M4	no	yes	no	ham

Perform below operations to know weather the M5 is spam or ham :

1. Count the occurrences of each class
2. Count the occurrences of each feature given the class
3. Calculate the prior probabilities
4. Calculate the likelihoods
5. Normalize probabilities

In [26]:
```
class_counts = messages_df['Class'].value_counts()


feature_counts_spam = messages_df[messages_df['Class'] == 'spam'].drop(columns='Class').apply(pd.value_counts).fillna(
feature_counts_ham = messages_df[messages_df['Class'] == 'ham'].drop(columns='Class').apply(pd.value_counts).fillna(0)


total_messages = len(messages_df)
prior_prob_spam = class_counts['spam'] / total_messages
prior_prob_ham = class_counts['ham'] / total_messages

# Given new message M5 = (Malicious = 'yes', Viagara = 'no', Meet = 'yes')
new_message = {'Malicious': 'yes', 'Viagara': 'no', 'Meet': 'yes'}


likelihood_spam = prior_prob_spam
likelihood_ham = prior_prob_ham

for feature, val in new_message.items():
    if val == 'yes':
        likelihood_spam *= (feature_counts_spam[feature]['yes'] / class_counts['spam'])
        likelihood_ham *= (feature_counts_ham[feature]['yes'] / class_counts['ham'])
    else:
        likelihood_spam *= ((class_counts['spam'] - feature_counts_spam[feature]['yes']) / class_counts['spam'])
        likelihood_ham *= ((class_counts['ham'] - feature_counts_ham[feature]['yes']) / class_counts['ham'])


evidence = likelihood_spam + likelihood_ham
posterior_prob_spam = likelihood_spam / evidence
posterior_prob_ham = likelihood_ham / evidence

print("Probability that the message is spam:", posterior_prob_spam)
print("Probability that the message is ham:", posterior_prob_ham)
```

```
Probability that the message is spam: 1.0
Probability that the message is ham: 0.0
```

From above observation we can say that the M5 is spam.