NAME: ____

AI2 Final Exam Fall 2025 Answer Key

Each question is worth 10 points.

1. There is about 0.05% chance of having a certain disease19. About 80% of the people with the disease experience symptoms of high fever and cough. About 5% of those that have symptoms end up testing positive. Joe wakes up one morning with a high fever and cough. What is the probability that Joe will test positive for disease19?

   ```
   about 5%, right there in question.
   ```

2. We run a bank's lending department. From past data, 15% of loans end up bad. Our loan application process is very detailed: *we record thousands of data points on each applicant*. After crunching past loans, we discover that bad loans were to people with mustaches 60% of the time (good loans involve a mustache only 10% of the time). A customer with a mustache walks in: use the Bayes rule to get probability that the customer will default on their loan:

   ```
   P(default) = 0.15, P(-default) = 0.85
   P(mustache|default) = 0.6, P(-mustache|defualt) = 0.4
   P(mustache|-default) = 0.1, p(-mustache|-default) = 0.9

   P(default|mustache) = P(mustache|default)*P(default) /
   (P(mustache|-default)*P(-default)+P(mustache|default)*P(default)) =
   (0.6 * 0.15) / (0.6 * 0.15 + 0.1*0.85) = 0.5143
   ```

3. We codify the rule from previous question, and put it into production. After sometime, we discover that the default rate remains mostly unchanged. What could have gone wrong?

   ```
   If we have thousands of features to choose from, some are bound to be
   correlated with the result we want by pure chance.
   ```

4. We next use income brackets: from past data, 70% of loan defaults are in $1-40k income bracket, 20% of loan defaults are in $40-100k income bracket, 10% are in $100-and-up bracket. 60% of good loans (non-defaulted) are from $1-40k income bracket, 35% from $40-100k bracket, and 5% from $100-and-up. A customer with $50k income walks in, according to Bayes rule what's the probability of default?

   ```
   P(default) = 0.15, P(-default) = 0.85
   P(50k|default) = 0.2, P(-50k|default) = 0.8
   P(50k|-default) = 0.35, P(-50k|-default) = 0.65

   P(default|50k) = P(50k|default)p(default) /
   ( P(50k|default)p(default) + P(50k|-default)p(-default) )
   (0.2*0.15) / (0.2*0.15 + 0.35*0.85) = 0.091603
   ```

5. Another feature that appears useful is whether the applicant has a car loan. We notice that 90% of defaulted applicants also had a car loan, while only 25% of non-default applicants had a car loan. A customer with a car loan walks in, according to Bayes rule what's the probability of default?

```
P(default) = 0.15, P(-default) = 0.85
P(car|default) = 0.9, P(-car|default) = 0.1
P(car|-default) = 0.25, P(car|-default) = 0.75

P(default|car) =P(car|defualt)*P(default) /
( P(car|defualt)*P(default) + P(car|-defualt)*P(-default) )
(0.9*0.15) / (0.9*0.15 + 0.25*0.85) = 0.3885
```

6. Being very clever, we first apply the income bracket model, followed by the car-loan check model. A customer with $50k income walks in, with an existing car loan, according to Bayes rule what's the probability of default?

```
P(default|car,50k) = P(car,50k|default)P(default)/P(car,50k)

Cannot be determined.  Not enough information.
Don't know P(car,50k|default).
```

7. Continuing from previous question, using Naive Bayes assumption, what's the probability of default after observing income bracket and car-loan feature?

```
Assuming P(car,50k|default) = P(car|default)*P(50k|default)
and P(car,50k) = P(car)P(50k)

P(default|car,50k) = P(car|default)*P(50k|default)*P(default)/
(P(car|default)*P(50k|default)*P(default)+P(car|-default)*P(50k|-default)*P(-defaul
P(default) = 0.15, P(-default) = 0.85
P(car|default) = 0.9, P(-car|default) = 0.1
P(car|-default) = 0.25, P(car|-default) = 0.75
P(50k|default) = 0.2, P(-50k|default) = 0.8
P(50k|-default) = 0.35, P(-50k|-default) = 0.65

(0.9*0.2*0.15) / (0.9*0.2*0.15 + 0.25*0.35*0.85) = 0.2663
```

8. Consider a 2-layer neural network with 2 inputs, 2 neurons in the hidden layer, and 1 output neuron. The network uses the ReLU activation function, defined as $\text{ReLU}(x) = \max(0, x)$. The network weights and biases are given as:

$$\text{Hidden layer weights: } W_1 = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\text{Output layer weights: } W_2 = \begin{bmatrix} 2 & 1 \end{bmatrix}, \quad b_2 = 2$$

The input to the network is:

$$x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Calculate the final output of the network. Show work (calculate the output of the hidden layer, apply ReLU, etc.)

```
import torch
w1 = torch.tensor([[2.0,1.0],[4.0,3.0]])
w2 = torch.tensor([2.0,1.0])
b1 = torch.tensor([2.0,1.0])
b2 = torch.tensor([2.0])
x = torch.tensor([2.0, 1.0])
o = (w2.matmul( (w1.matmul(x) + b1).relu() ) + b2).relu()
28
```

9. Fair coin flipping game: We start with \$1. Heads we win 50%, tails we lose 50%. After 3 rounds, with a fair coin, the *arithmetic mean* value we will have:

   `$1.00`

10. Fair coin flipping game. We start with \$1. Heads we win 50%, tails we lose 50%. After 3 rounds, with a fair coin, the *median* value we will have:

    `$0.75`