Question 1

(a) Suppose we have $n$ balls placed in $n$ bins, each numbered from 1 to $n$. Let each ball have an equal chance of going into any of the bins 1, 2, ..., $n$. What is the probability that bin $i$ is empty?

(b) What is the expected number of empty bins?

(c) Suppose we have a bin containing $b$ blue balls and $r$ red balls. We draw $n$ balls, without replacement, from the bin. What is the probability that the $i$th blue ball was picked after $n$ draws?

(d) Find the expected number of blue balls drawn.

Answer 1
Question 2

We play a game and we start with $0. Every turn, we flip a regular coin. If we get heads, we win 4 dollars. Otherwise, we lose all of our money. We can stop flipping at any time and keep our winnings, but if we get a total of 2 tails, then we lose everything and the game ends. Let’s figure out how to win as much money as possible from this game.

(a) Given that we have \( M \) dollars so far and one tail, what is the expected value of a flip?

(b) Given that we have \( M \) dollars so far and no tails, what is the expected value of a flip? (Hint: apply part (a). When do we flip again?)

(c) What is the optimal strategy for our game, and how much money do we expect to win?

(d) Consider the following, different game. We start with 2 dollars. We flip a regular coin at every turn. If we get heads, then we double our money. Otherwise, the game ends and we lose everything. How much money do we expect to win from this game? Would you pay this amount to play?

Answer 2
Question 3

Alice, Bob, and Carol play a chess tournament. The first game is played between Alice and Bob. Each subsequent game is played between the winner of the previous game, and the person who did not play in the previous game. The tournament ends when some player wins two games in a row. We can write a tournament history as the list of game winners, so for example ACBAA corresponds to the tournament where Alice won games 1, 4, and 5, Carol won game 2, and Bob won game 3.

(a) Draw a tree structure of all possible outcomes of the first several games of the tournament. Taking a picture of handwritten diagram is acceptable. Include enough rounds to show four scenarios where Carol wins.

(b) If each player has an equal chance of winning each game they play, what is the expected number of games to determine a winner?

(c) If each player has an equal chance of winning each game they play, what is the probability that each player wins the tournament?

   *Hint:* You may find it useful to look into the summation of an infinite geometric series.

(d) Again assume each player has an equal chance of winning each game they play. Given that Carol wins at least one game, what is the probability each player wins the tournament?

Answer 3
Question 4

As a faculty member at Brown University, Eli receives a lot of email, and he is in desperate need of a method for separating important messages from spam. Now that you have learned some discrete probability, could you help him out?

Our dataset contains emails with labels of spam or ham. You’ll use a “naive Bayes” classifier to identify spam emails. The vocabulary vocab consists of W words, where each character string has been mapped to a distinct integer index. The training data matrix trainFeat is a $D \times W$ matrix, where each row represents an email and each column indicates whether that word appears in that email at least once. The ground truth labels are stored in trainLabels, with “1” indicating spam and “0” ham. Test data, to be used for evaluating performance but not estimating probabilities, is stored in testFeat, testLabels.

To define a probabilistic model of this data, we let $Y_i = S$ if email $i$ is spam and $Y_i = H$ if email $i$ is ham (not spam). We assume that the two classes are equally likely a priori:

$$P(Y_i = S) = P(Y_i = H) = 0.5$$

(1)

To encode the data that will be used for classification, we let $X_{ij} = 1$ if email $i$ contains an instance of word $j$, and $X_{ij} = 0$ if email $i$ does not contain the word $j$. The set of all available data about email $i$ is then $X_i = \{X_{ij} \mid j \in 1, ..., W\}$.

To implement a simple Bayesian classifier, we will compute the posterior distribution $P(Y_i \mid X_i)$ of the class label given the observed features. If $P(Y_i = S \mid X_i) > P(Y_i = H \mid X_i)$, we classify email $i$ as spam; otherwise, we classify it as ham.

(a) Either prove or disprove the following statement: the Bayesian classifier described above is equivalent to a classifier that assigns label spam if $P(X_i \mid Y_i = S) > P(X_i \mid Y_i = H)$, and label ham otherwise. Base your reasoning on Equation 1 and Bayes’ rule.

To further simplify the modeling problem, a naive Bayes classifier assumes that given the class label, the observed word features are conditionally independent. From the definition of independence, this implies that

$$P(X_{i,:} \mid Y_i = S) = \prod_{j=1}^{W} P(X_{ij} \mid Y_i = S), \quad P(X_{i,:} \mid Y_i = H) = \prod_{j=1}^{W} P(X_{ij} \mid Y_i = H)$$

(2)

(b) A simple way to estimate the probabilities above is by counting how many times each event occurs in the training data. Let $N_s$ be the total number of spam emails, $N_{sj}$ the number of spam emails in which word $j$ occurs, $n_h$ the total number of ham emails, and $N_{hj}$ the number of ham emails in which word $j$ occurs. we then set

$$P(X_{ij} \mid Y_i = S) = \frac{N_{sj}}{N_s}, \quad P(X_{ij} = 0 \mid Y_i = S) = 1 - P(X_{ij} = 1 \mid Y_i = S) = \frac{N_s - N_{sj}}{N_s}$$

$$P(X_{ij} \mid Y_i = H) = \frac{N_{hj}}{N_h}, \quad P(X_{ij} = 0 \mid Y_i = H) = 1 - P(X_{ij} = 1 \mid Y_i = H) = \frac{N_h - N_{hj}}{N_h}$$

Write Matlab code to compute these probabilities using data in trainFeat, trainLabels.

(c) Consider a simplified dataset that only contains the presence or absence of a single word, $j$ = “money”. Compute and report the numerical values of the conditional probabilities $P(X_{ij} = 1 \mid Y_i = S)$, $P(X_{ij} = 1 \mid Y_i = H)$. What is the test accuracy of a Bayesian classifier based on this single word?
(d) Repeat part (c) for a different single word, \( j = \text{"thanks"} \). Provide an intuitive explanation for any differences in classification performance.

(e) Consider a slightly larger dataset which contains the presence or absence of the two words (“money,thanks”) from parts (c-d). Using the naive Bayes assumption from Equation (2), determine the test accuracy of a classifier based on these two words.

When the number of words \( W \) is large, the probabilities of Equation (2) become very small, and can underflow to 0 when using finite-precision arithmetic on a computer. To avoid this, we will instead work in the log-domain, and pick the class whose log-probability is largest. Because \( \log(ab) = \log(a) + \log(b) \), we have:

\[
P(X_i, | Y_i = S) = \log \prod_{j=1}^{W} P(X_{ij} | Y_i = S) = \sum_{j=1}^{W} \log P(X_{ij} | Y_i = S) \tag{3}
\]

with a similar identity for \( \log P(X_i, | Y_i = H) \).

(f) Using the identity in Eq. (3), modify your classification code to compute the log-probability of the spam and ham classes in a numerically robust fashion. Determine the test accuracy of a classifier based on all \( W \) words in the full dataset. Hint: this classifier should take seconds (not minutes) to train and test, and be much more accurate than part (f).

(g) Suppose we relax the assumption \( P(Y_i = S) = P(Y_i = H) = 0.5 \). How would you estimate these probabilities, and how would your classifier need to change to take this into account?

(h) Suppose a word never appears in the training set (i.e. there exists some \( j \) such that \( X_{i,j} = 0 \) for all \( i \)). What would your classifier predict for documents that contain this word? Is this reasonable? If not, propose what you would like the classifier to do in this situation, and suggest a method to obtain the desired behavior.

(i) Consider an output \( Y \) and inputs \( X_1 \) and \( X_2 \) where \( Y \) represents the exclusive or operation between \( X_1 \) and \( X_2 \), namely:

\[
Y_1 = \begin{cases} 
1 & X_1 = 0 \text{ and } X_2 = 1 \\
1 & X_1 = 1 \text{ and } X_2 = 0 \\
0 & \text{otherwise}
\end{cases}
\]

Now, consider the following dataset:

<table>
<thead>
<tr>
<th>Y</th>
<th>X_1</th>
<th>X_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

What will a Naive Bayesian algorithm trained on this data set classify the following points as? You do not need to show your math for this step.

(a) \((0,0)\)
(b) (0,1)
(c) (1,0)
(d) (1,1)

Why does the algorithm incorrectly label some of these points? Would increasing the size of the training data solve this problem? Are there any assumptions implicit in the naive Bayes model that are inconsistent with an exclusive or function? Are there types of data for which a naive Bayesian classifier would not be a good choice?

Note that this is an open-ended question.

**Answer 4**