

CSE 312 – Problem Set 3

Due Wednesday, April 22, 11:59pm

Spring 2026

Instructions

For details on expectations for solutions, collaboration policy, late policy, etc, see the instructions on the prior problem set.

Solutions submission. You must submit your solution via Gradescope. In particular:

- Problem 1 will be done on Gradescope.
- Submit the solutions to problems 2-5 under “PSet 3 [Written]” as on previous problem sets.
- For the programming part (Problem 6), submit your code under “PSet 3 [Coding]” as a file called `cse312_pset3_nb.py`.
- Do not submit a solution to problem 7. That is just for fun.

1 Gradescope Problems (30 points)

Complete this problem on [gradescope](#). Please check your answers as it will be autograded so we cannot give partial credit.

2 Miscounting (10 points)

Consider the question: what is the probability of getting a **7-card** poker hand (order doesn't matter) that contains at least two 3-of-a-kind (3-of-a-kind means three cards of the same rank). For example, this would be a valid hand: ace of hearts, ace of diamonds, ace of spades, 7 of clubs, 7 of spades, 7 of hearts and queen of clubs. (Note that a hand consisting of all 4 aces and three of the 7s is also valid.)

Here is how we might compute this:

Each of the $\binom{52}{7}$ hands is equally likely. Let E be the event that the hand selected contains at least two 3-of-a-kinds. Then

$$\mathbb{P}(E) = \frac{|E|}{\binom{52}{7}}$$

To compute $|E|$, apply the product rule. First pick two ranks that have a 3-of-a-kind (e.g. ace and 7 in the example above). For the lower rank of these, pick the suits of the three cards. Then for the higher rank of these, pick the suits of the three cards. Then out of the remaining $52 - 6 = 46$ cards, pick one. Therefore

$$|E| = \binom{13}{2} \cdot \binom{4}{3} \cdot \binom{4}{3} \cdot \binom{46}{1} \quad \text{and hence} \quad \mathbb{P}(E) = \frac{\binom{13}{2} \cdot 4^2 \cdot 46}{\binom{52}{7}}.$$

Explain what is wrong with this solution. If there is over-counting in $|E|$, characterize all hands that are counted more than once, and how many times each such hand is counted. If there is under-counting in $|E|$, explain which hands are not counted. Also, **give the correct answer for $\mathbb{P}(E)$.**

3 You knew it all along! (10 points)

You are taking a multiple choice test that has 4 answer choices for each task. In answering a task on this test, the probability you know the correct answer is p . If you know the correct answer, you will definitely select it. If you don't know the correct answer, you choose one (uniformly) at random. What is the probability that you knew the correct answer to a task, given that you answered it correctly? Use Bayes Theorem, and use the following names for the relevant events, e.g. let K be the event that you know the correct answer and let C be the event that you answer the task correctly (whether you knew the answer or not).

4 Doggone, Doggtwo, Doggthree...(10 points)

A hunter has two hunting dogs. One day, on the trail of some animal, the hunter comes to a place where the road diverges into two paths. She knows that each dog, independent of the other, will choose the correct path with probability p . The hunter decides to let each dog choose a path, and if they agree, take that one, and if they disagree, to randomly pick a path. What is the probability that she ends up taking the correct path? Hint: Use the law of total probability, partitioning based on whether the dogs choose the same path or different paths.

5 Aces (12 points)

Suppose that an ordinary deck of 52 cards (which contains 4 aces) is randomly divided into 4 hands of 13 cards each. We are interested in determining p , the probability that each hand has an ace. Let E_i be the event that the i -th hand has exactly one ace. Our goal is to determine

$$p = \mathbb{P}(E_1 \cap E_2 \cap E_3 \cap E_4)$$

using the chain rule.

- (3 points) What is $\mathbb{P}[E_1]$, the probability that our first hand has exactly one ace?
- (3 points) Given that E_1 has occurred, there are now 39 cards left in the deck, with 3 of them aces. What is $\mathbb{P}[E_2 | E_1]$?
- (6 points) Continue this argument for $\mathbb{P}[E_3 | E_1, E_2]$ and $\mathbb{P}[E_4 | E_1, E_2, E_3]$, and use the chain rule to compute p .

6 Naive Bayes [Coding] (25 points)

Use the Naive Bayes Classifier to implement a spam filter that learns word spam probabilities from our pre-labeled training data and then predicts the label (ham or spam) of a set of emails that it hasn't seen before.

For an introduction to the Naive Bayes Classifier and details on implementation, see the video and slides (as well as some self-check questions) linked at [this edstem lesson](#). Also read Section 9.3 from [the book](#). To solve the task, we have set up an [edstem lesson](#). In particular, write your code to implement the functions `fit` and `predict` in the provided file, `cse312_pset3_nb.py`.

You will be able to run your code directly within edstem, and to test it, using the “Mark” option. This, however, will not evaluate your solution. Instead, once you’re ready to submit, you can right-click the files in the directory to download them. Please upload your completed `cse312_pset3_nb.py` to Gradescope under “PSet3 [Coding]”. Similarly for

Some notes and advice:

- Read about how to avoid floating point underflow using the log-trick as described in these [notes](#).
- Make sure you understand how Laplace smoothing works.
- Remember to remove any debug statements that you are printing to the output.
- **Do not directly manipulate file paths or use hardcoded file paths.** A file path you have hardcoded into your program that works on your computer won’t work on the computer we use to test your program.
- Needless to say, you should practice what you’ve learned in other courses: document your program, use good variable names, keep your code clean and straightforward, etc. Include comments outlining what your program does and how. We will not spend time trying to decipher obscure, contorted code. Your score on Gradescope is your final score, as you have unlimited attempts.
- We will evaluate your code on data you don’t have access to, in addition to the data you are given.
- Remember, it is not expected that Naive Bayes will classify every single test email correctly, but it should certainly do better than random chance! As this algorithm is deterministic, you should get a certain specific test accuracy around 90-95%, which we will be testing for to ensure your algorithm is correct. Note that we will run your code on a test dataset you haven’t seen, but you will know immediately if you got full score.
- **Start early!**

7 For Fun Problem (will not be graded)

This is a not-too-difficult problem for those of you that would enjoy proving a cool probabilistic fact. You will not be turning anything in for this.

You are shown two envelopes and told the following facts:

- Each envelope has some number of dollars in it, but you don’t know how many.
- The amount in the first envelope is different from the amount in the second.
- Although you don’t know exactly how much money is in each envelope, you are told that it is an integer number of dollars that is at least 1 and at most 100.

- You are told that you can pick an envelope, look inside, and then you will be given a one-time option to switch envelopes (without looking inside the new envelope). You will then be allowed to keep the money in envelope you end up with.

Your strategy is the following:

- a) You pick an envelope uniformly at random.
- b) You open it and count the amount of money inside. Say the result is x .
- c) You then select an integer y between 1 and 100 uniformly at random.
- d) If $y > x$, you switch envelopes, otherwise you stay with the envelope you picked in step (a)

Show that you have a better than 50-50 chance of taking home the envelope with the larger amount of money in it. More specifically, suppose the two envelopes have i and j dollars in them respectively, where $i < j$. Calculate the probability that you take home the envelope with the larger amount of money.