USC CSCI 467 Intro to Machine Learning	**Assignment 0** Due: January 19, 2023, 11:59pm	Spring 2023 Instructor: Robin Jia

This assignment is primarily designed to remind you of various prerequisites that will be important for this class. If you look at these problems and find that you don't remember how to do everything, that is expected. If you look at these problems and find the concepts to be unfamiliar to you, there may be prerequisites that you should consider taking before you enroll in this class.

Instructions: Submit a pdf document on Gradescope that includes **both your responses to all questions and your code for Question 4**. You are highly encouraged to use LaTeX to typeset your submission. If you are using LaTeX, we recommend using the `minted` library (https://www.overleaf.com/learn/latex/Code_Highlighting_with_minted) to add syntax highlighting to your code.

This assignment has 4 questions, for a total of 50 points.

Question 1: Probability (16 points)

(a) (Bayes Rule) A company develops a test for COVID. If a patient has COVID, the test returns positive with probability .75. If a patient does not have COVID, the test returns positive with probability .01. Suppose that 2% of the population is currently infected with COVID. Compute:

i. (3 points) $P(\text{Patient has covid} \mid \text{Test is positive})$

ii. (3 points) $P(\text{Patient has covid} \mid \text{Test is negative})$

(b) (Random variables, independence, and expected value) Two classmates, Alice and Bob, are about to take an exam for the same class. Let A be the random variable denoting Alice's exam score (between 0 and 100), and let B be the random variable denoting Bob's exam score.

i. (4 points) Are A and B independent random variables? Justify your reasoning in 1-2 sentences.

ii. (3 points) Suppose you knew that $P(A \geq 90) = 0.6$ and $P(B \geq 90) = 0.3$. If A and B are independent, compute $P(A \geq 90 \text{ and } B \geq 90)$. If A and B are not independent, explain why you do not have enough information to compute $P(A \geq 90 \text{ and } B \geq 90)$.

iii. (3 points) Suppose you knew that $\mathbb{E}[A] = 92$ and $\mathbb{E}[B] = 84$. Do you have enough information to compute $\mathbb{E}[A + B]$? If so, compute it. Either way, explain your reasoning.

Question 2: Single-variable Calculus (10 points)

You find a mysterious coin on the ground and flip it five times. You observe the sequence Heads, Tails, Heads, Heads, Heads.

(a) (2 points) Let p be the (unknown) probability that the coin lands heads when flipped. Write an expression for $L(p)$, the probability of the observed data as a function of p.

(b) (6 points) What value of p maximizes $L(p)$? (Hint: Consider taking the derivative of $\log L(p)$. Also remember to apply the second derivative test to confirm that you have found a local maximum and not a local minimum.)

(c) (2 points) Provide an intuitive explanation for this optimal value of p.

Question 3: Linear Algebra (12 points)

(a) (Vector geometry) Let $v = [1, -3, 5]$ and $w = [-2, 1, 2] \in \mathbb{R}^3$. Compute:

  i. (2 points) The unit vector u in the same direction as w.

  ii. (2 points) The (vector) component of v that is parallel to w.

  iii. (2 points) The (vector) component of v that is perpendicular to w. Verify that this vector is in fact perpendicular to w.

(b) (6 points) A positive semi-definite matrix is a matrix $A \in \mathbb{R}^{d \times d}$ such that for all vectors $x \in \mathbb{R}^d$, $x^\top A x \geq 0$. Prove that for any vector $u \in \mathbb{R}^d$, the matrix $uu^\top$ is positive semi-definite. (Hint: While you can show this by taking an eigendecomposition, there is a simpler proof.)

Question 4: Programming and Gradient Descent (12 points)

This problem will be a little different from the rest, as it is not about prerequisites but instead about gradients and doing linear algebra in numpy. Completing this problem will also ensure that you have a suitable python environment installed. You'll be ready to start this after Lecture 2.

numpy. numpy is a widely used python library for numerical linear algebra. We do not expect you to already be familiar with numpy; however, you should be comfortable writing python code. The advantage of using numpy, as opposed to writing linear algebra operations like matrix multiplication in raw python, is efficiency. Under the hood, numpy is implemented in C and makes use of fast linear algebra libraries.

Setup. We require python3 for this class. The assignments were developed with python 3.8.0 and numpy 1.23.4; other versions may work but we recommend using the same or similar versions if possible. You can install numpy by following these instructions: `https://numpy.org/install/`. Once you have installed numpy, download the starter code file called `hw0.zip`.

Problem. In this problem we will implement gradient descent in numpy to find the minimum of a function from $\mathbb{R}^2 \to \mathbb{R}$. In particular, let $v_1 = [1, 4]$ and $v_2 = [1, 0]$. Suppose we want to find a vector that has large dot product with v_1 but also is close to v_2. We can formulate this as an optimization problem: define

$$f(x) = -v_1^\top x + \|x - v_2\|^2.$$

We can now achieve our goal by minimizing f. Note the negative sign in front of $v_1^\top x$; because we want to maximize $v_1^\top x$, we want to minimize its negation. Here, $\|w\|$ denotes the Euclidean norm.

(a) (2 points) In `q4_numpy.py`, implement the function `f(x)`. Your solution should **not** use any for loops. You may find the numpy function `numpy.sum()`, the `.dot()` method, and the `**` operators useful.

(b) (4 points) Now, compute the mathematical expression for $\nabla_x f(x)$.

(c) (2 points) In `q4_numpy.py`, implement the function `grad_f(x)`, which should return the expression you wrote in the previous part. Again, your solution should **not** use any for loops.

(d) (4 points) Finally, in `q4_numpy.py`, implement the function `find_optimum()`. You should run gradient descent with the specified default learning rate and number of iterations. (You will need to use a single for loop for this.) Start at the initial value of $x = [0, 0]$. (You can use the `numpy.zeros()` method for this.) **Report the final value of x and $f(x)$ that you find.**