Problem 1  QII measures (10 points)

Decision-making systems that use machine learning are increasingly used to aid decision making in today’s society. Such decisions could be about online personalization or credit and insurance decisions. Unfortunately, it is often difficult to explain why a certain decision was made. Datta et al 2016 introduce a family of Quantitative Input Influence (QII) measures to capture the degree of influence of inputs on outputs of systems that can explain decisions about individuals (e.g., a loan decision) and groups (e.g., disparate impact based on gender). Because a single input may not always have a high influence, QII measures also quantify the joint influence of a set of inputs (e.g., age and income) on outcomes (e.g., loan decisions), and the marginal influence of an individual input within such a set (e.g., income).

An implementation of the QII measures over several data sets is available at [https://github.com/cmu-transparency/tool-qii](https://github.com/cmu-transparency/tool-qii). Clone the repository and consult the instructions in the README.md to run the tool.

Exercise 1. Use the tool to investigate the provided nlsy97 dataset.

(a) Display a figure of the QII on outcomes (that doesn’t require specifying a measure). Which attribute is the most influential? Include the command you invoked and the generated figure.

(b) Display the figure of the QII on group disparity (measure dicrim). Which attribute has the greatest negative impact on outcomes? Include the command you invoked and the generated figure.
Problem 2 Anonymous Communication (35 points)

Exercise 2. General Protocol (15 points) In the lecture, we discussed the Dining Cryptographers protocol. In this problem, we will explore how to use that protocol as a building block to construct a general protocol for anonymous communication. Consider a group of $n$ agents. (You may want to read The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability\(^1\))

(a) Describe a protocol using which one of the $n$ agents can send an $m$-bit message. Explain informally why the protocol is correct (i.e., all agents receive exactly the message that was sent) and anonymous (i.e., none of the other agents have any clue who the real sender is).

(b) State and prove rigorously that anonymity is preserved by the protocol for the case where $n = 4$ and $m = 1$. (You need to show that from the point of view of any non-sender, the probability of any of the other agents being the sender is $1/3$).

(c) How many bits of randomness and how many message transmissions are needed to complete this protocol with $n$ agents and an $m$-bit message?

(d) How robust is this protocol to collusion, i.e., if $k$ out of the $n$ non-sender agents collude, what is the probability that they can figure out who the real sender is?

\(^1\)Available at http://users.ece.cmu.edu/~adrian/731-sp04/readings/dcnets.html.
Exercise 3. Hidden services (5 points) Tor can also provide anonymity for servers, apart from providing anonymity for clients. Read the relevant part of the paper *Tor: The Second-Generation Onion Router*[^3] and explain how hidden services work in Tor. The explanation **must be a bulleted list of the main points.** Marks will be deducted for writing paragraphs.

- ...

Exercise 4. Nymble (15 points) Tor can sometimes lead to some undesirable consequences. Look at the paper *Nymble: Anonymous IP-Address Blocking*[^4] and answer the following questions:

(a) What potential problem with Tor are identified in the paper?

(b) Provide an overview of how the Nymble system works. Section 3 in the paper has such an overview. You can read that overview, however, your answer must be in your own words.

(c) List the (informally) cryptographic properties that the Nymble system relies on.

Problem 3 Secure Multiparty Computation (28 points)

Suppose Alice has two pieces of information, and Bob wishes to learn one of them, but Alice doesn’t want him to learn both, and Bob doesn’t want Alice to learn which piece he requested. They may

accomplish this by means of a technique known as oblivious transfer. In class, we demonstrated an implementation of this done with playing cards (hereafter referred to as Playing Card Oblivious Transfer):

1. Alice’s two pieces of information may be expressed by the value of two cards, one red and one black, and Bob is interested in learning the value of either the red card or the black card.
2. Alice places her two cards face-down and tells Bob which one is red and which one is black.
3. Alice turns her back, and Bob peeks just at the card of the color he is interested in learning about.
4. Bob pushes the cards into a pile to destroy evidence of his peeking.

Because face-down cards all look the same and Bob only looks at the face of one of the cards, he only learns one of the two pieces of information that Alice has available. Because Alice turns her back while Bob looks, and Bob then moves all the cards so that Alice cannot see which one he touched, Alice cannot know which piece of information Bob learned.

Exercise 5. In this problem, we will examine some limitations and extensions of this protocol.

(a) (2 points) Playing Card Oblivious Transfer is secure only against semi-honest behavior on the part of the participants; that is, its guarantees of secrecy only holds if both parties follow the protocol. Give one example each of how Alice and Bob could violate the protocol to learn information they shouldn’t or to prevent the other from learning information they should.

(b) (3 points) Suppose we relax the condition that says that Alice must not learn which piece of information Bob learns. Can you design a protocol based on Playing Card Oblivious Transfer which would allow Bob to learn exactly one card and which would work even if Bob did try to cheat?

Alice and Bob would like to do more than just share single pieces of information. Ideally, they’d like to be able to do computations where each of them has some of the inputs so that they both learn the output, but neither of them learns anything about the other’s inputs beyond that which follows from the output. Alice and Bob may be considered semi-honest: they will follow the protocol you describe.
(c) (5 points) The AND playing card logic gate takes two input cards and returns Red if both inputs are Red and Black otherwise. Design a protocol based on Playing Card Oblivious Transfer that allows Alice and Bob to check if both their cards are Red without sharing any other information. They are assumed to each provide one card as input and the color of the cards is no longer drawn on their backs.

(d) (5 points) If both inputs to the AND playing card logic gate are Red, Alice and Bob will each learn from the result also being Red that the other person’s card is Red. Why does this not violate our preference that Alice and Bob learn only the output of the gate? In your protocol, what does Alice learn about Bob’s card if her card is Black?

(e) (3 points) The OR playing card logic gate takes two input cards and returns Red if either input is Red, and Black otherwise. Design a protocol based on Playing Card Oblivious Transfer that allows Alice and Bob to check if either of their cards are Red without sharing any other information.

Alice and Bob are very happy with the logic gates you gave them. But now suppose Alice has two cards, x and y, and Bob has one card, z. Alice and Bob would like to evaluate the playing card logic gate sequence ((x AND z) OR y) such that they each learn the final result of this computation, but no other information about the other person’s cards.

(f) (2 points) Give a table providing the output of this playing card logic gate sequence for all possible colors of x, y, and z.

(g) (8 points) Alice and Bob first attempt to do this by using the secure playing card logic gate protocol you found in question (c) for AND, then taking the result of that and running it
through the protocol you found in question (e). Why is this method not as private as they would like it to be?

(h) (10 BONUS points) Design a protocol for Alice and Bob to evaluate multiple playing card logic gates so that they only learn the final result of the computation and nothing else about each others’ cards. Slides 38-44 of these lecture notes [http://zoo.cs.yale.edu/classes/cs467/2017f/lectures/ln24.pdf](http://zoo.cs.yale.edu/classes/cs467/2017f/lectures/ln24.pdf) may prove helpful.