CS640 Homework 1: ROC Analysis and Ethics

In this assignment, you will answer some questions regarding ROC analysis.

Collaboration

You are allowed to work in a team of at most **three** on the coding part (**Q1**), but you must run the experiments and answer written questions independently.

Instructions

General Instructions

In an ipython notebook, to run code in a cell or to render Markdown+LaTeX press `Ctrl+Enter` or `[>|]` (like "play") button above. To edit any code or text cell (double) click on its content. To change cell type, choose "Markdown" or "Code" in the drop-down menu above.

Most of the written questions are followed by a cell for you enter your answers. If not, please insert one. Your answers and the questions should **not** be in the same cell.

Instructions on Math

Some questions require you to enter math expressions. To enter your solutions, put down your derivations into the corresponding cells below using LaTeX. Show all steps when proving statements. If you are not familiar with LaTeX, you should look at some tutorials and at the examples listed below between $..$.

Alternatively, you can scan your work from paper and insert the image(s) in a text cell.

Submission

Once you are ready, save the notebook as PDF file (File -> Print -> Save as PDF) and submit to Gradescope. Make sure all outputs are visible. If you encounter rendering issue, try changing the page setup or consider taking screenshots.

Q0: Name(s)

Please write your name in the next cell. If you are collaborating with someone, please list their names as well.

Double-click (or enter) to edit

Q1: Confusion Matrix

In the next cell, write code to manually compute the confusion matrix for a **binary** classification task and run the cell to test your code.

The function `confusion_matrix` from sklearn is used to verify your answers, but you should **not** use it in your implemention.

```
import numpy as np
from sklearn.metrics import confusion_matrix

def compute_confusion_matrix(YTrue, YPredict):
    """
    Computes the confusion matrix as a numpy matrix. For convention, the
    vertical axis represent YTrue while the horizontal axis represent YPredict.
    """
    cm = np.zeros((2, 2))

    ############################## start of your code ##############################
    for i in range(len(YTrue)):
        cm[YTrue[i], YPredict[i]] += 1
    ############################### end of your code ###############################

    return cm

is_correct = True
for _ in range(10):
    rng = np.random.default_rng()
    YTrue = rng.choice([0, 1], size = 100)
    YPredict = rng.choice([0, 1], size = 100)
    if (compute_confusion_matrix(YTrue, YPredict) != confusion_matrix(YTrue, YPredict)).a
        is_correct = False
        break
print(is_correct)
```

```
True
```

Q2: Metrics

Q2.1

Suppose that we are given the confusion matrices and the ROC curves of two models **A** and **B**.

		Predicted class	
		0	1
Actual class	0	30	18
	1	26	26

(A)

		Predicted class	
		0	1
Actual class	0	16	39
	1	31	14

(B)

For each of the two models,

1. compute true positive rate (TPR), false negative rate (FNR), false positive rate (FPR), and true negative rate (TNR); and
2. compute accuracy, precision, recall and F-1 score.

You can do it either by hand or by code (do **not** use existing functions). Either way, show details of your calculations in a new cell using variables TP, FN, FP, and TN. Round your answers to four digits if you choose to use decimals.

```
 1 import numpy as np
 2
 3 cm_A = np.matrix([[30, 18], [26, 26]])
 4 cm_B = np.matrix([[16, 39], [31, 14]])
 5 cms = {"A" : cm_A, "B" : cm_B}
 6
 7 for name in cms:
 8     cm = cms[name]
 9     TP, FN, FP, TN, P, N = cm[1, 1], cm[1, 0], cm[0, 1], cm[0, 0], cm[1, :].sum(), cm[0,
10     TPR = TP / P
11     FNR = FN / P
12     FPR = FP / N
13     TNR = TN / N
14     accuracy = (cm[0, 0] + cm[1, 1]) / cm.sum()
15     precision = TP / (TP + FP)
16     recall = TPR
17     f1 = 2 * precision * recall / (precision + recall)
```

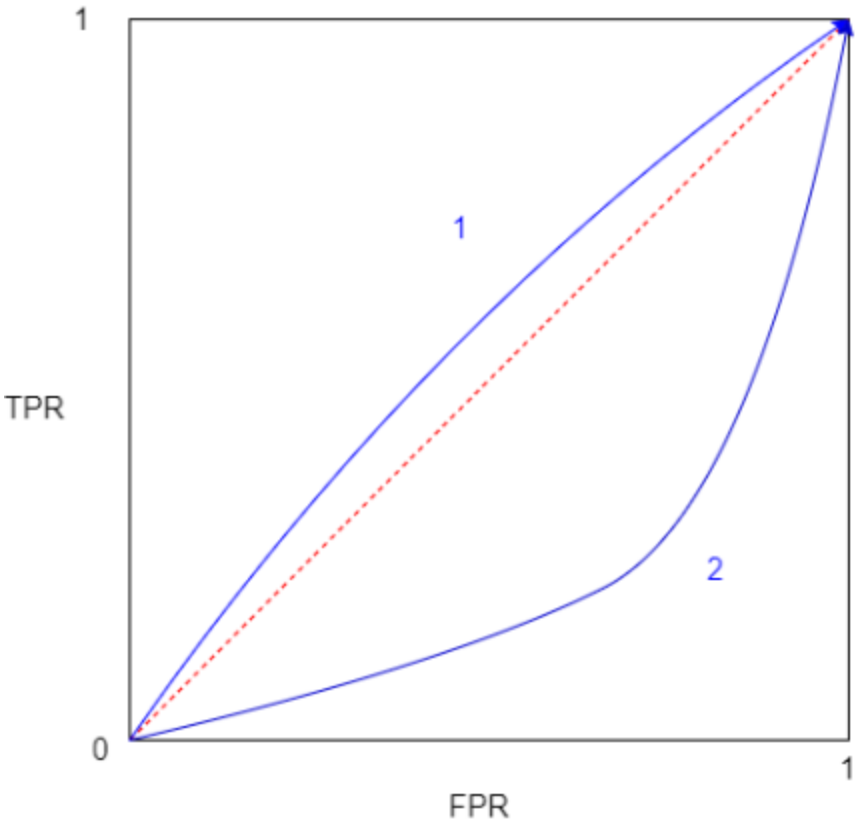
```
    print("Model " + name)
    print("TPR = " + str(np.round(TPR, 4)))
    print("FNR = " + str(np.round(FNR, 4)))
    print("FPR = " + str(np.round(FPR, 4)))
    print("TNR = " + str(np.round(TNR, 4)))
    print("accuracy = " + str(np.round(accuracy, 4)))
    print("precision = " + str(np.round(precision, 4)))
    print("recall = " + str(np.round(recall, 4)))
    print("f1 = " + str(np.round(f1, 4)))
    print()
```

```
Model A
TPR = 0.5
FNR = 0.5
FPR = 0.375
TNR = 0.625
accuracy = 0.56
precision = 0.5909
recall = 0.5
f1 = 0.5417

Model B
TPR = 0.3111
FNR = 0.6889
FPR = 0.7091
TNR = 0.2909
accuracy = 0.3
precision = 0.2642
recall = 0.3111
f1 = 0.2857
```

Q2.2

The two ROC curves are labeled as 1 and 2 in the following figure.



1. Supposed that the model predictions are final (i.e., they cannot be flipped). Which one represents a better model? Briefly explain.
2. Based on your results in Q2.1, which ROC curve is more likely to be associated with model **A** and which is more likely to be associated with model **B**? Briefly explain.

Answer

1. Curve 1 represents a better model. Because curve 1 lies above the diagonal line while curve 2 lies below the diagonal line, curve 1 is closer to the ideal model while curve 2 is closer to the worst model.
2. For model A, (FPR, TPR) = (0.375, 0.5) falls in the upper triangle. For model B, (FPR, TPR) = (0.7091, 0.3111) falls in the lower triangle. Therefore, model A is more likely to be associaed with curve 1 while model B is more likely to be associated with curve 2.

Q3: Threshold

Now suppose that we have a state-of-the-art model to predict earthquakes (negative vs positive) and by default the threshold value is set to 0.5. If we don't want to trigger too many false alarms, how should we tune the threshold value (i.e., should we increase it or decrease it)? Briefly explain.

Answer

The threshold value needs to be **increased**. Less false alarm means the model needs a higher confidence level (i.e., threshold) to show positive predictions (or a higher precision and a lower recall).

Q4: AI and Ethics

For each the following three questions, provide an answer with **no more than three** sentences.

Q4.1

Watch Joy Buolamwini's TED talk on How I'm fighting bias in algorithms. Joy mentions that judges use machine-generated risk scores to determine how long an individual is going to spend in prison. The state of Wisconsin uses the AI system Northpointe for making sentencing and parole decisions. Research this issue on the internet and provide one link to material you have read.

Double-click (or enter) to edit

Q4.2

Use some the bias terminology of Suresh and Guttag, 2021 (see lecture slides), to discuss whether you think the Northpointe system is a responsible use of AI.

Double-click (or enter) to edit

Q4.3

Research at BU and MIT by Canetti et al. (arxiv.org/pdf/1810.02003.pdf) considers the scenario that decisions whether a prisoner should stay or leave prison can be deferred. In the publicly-available Northpointe data, decisions about Caucasian prisoners are deferred for 9% of prisoners, while decisions about African-Americans are deferred for 20% of the prisoners. Canetti et al. propose "equalizing methods" to change the AI system so that deferral decisions would be made at about the same rate for both racial groups. Briefly discuss what you think about adding an equalizing method to Northpointe's system. Would this yield a fair AI system? You may refer to Suresh and Guttag's terminology.

Double-click (or enter) to edit

Q5: Turing Test

The Turing test is an imitation game. Argue the point that we should not aim for imitating human intelligence - artificial intelligence may be different from human intelligence (3-5 sentences only).

Double-click (or enter) to edit

Q6: Sentence Inference

For the topic of AI and Ethics, design example sentences that show the concept of inference. In particular, provide sentences A, B1, B2 and B3 such that 1) A entails B1, 2) A contradicts B2, and 3) A and B3 are neutral (independent statements).

Sentence A:

Sentence B1:

Sentence B2:

Sentence B3: