Activity08: Understanding Classification Error

Goal

In this activity you will practice calculating the ROC curve and computing the confusion matrix.

Instructions

You NEED to hand in your solution for this Activity**,** please see the due date on Canvas.

Solution key to the questions will be released after the deadline.

Submission format: Please upload a single PDF file of this document after inserting your answers in the space after each question, which shows all the steps you took, intermediate calculations, and the code you wrote and used.

You can upload extra .m files as supplementary document in a .zip file beside the pdf, but it's not going to add any extra grade.

I. Evaluate an AI-based COVID-19 Diagnosis System

Note: The information provided here is based on real research, however, given the seriousness of COVID19, please assume the information here is hypothetical and may contain errors and should not be used for any purpose beyond this assignment.

The current gold-standard for diagnosis of COVID-19 is real-time polymerase chain reaction (RT-PCR) lab test [Bai 2020]. However, lab resources are expensive, limited and time consuming. A quick, cheaper and non-invasive alternative may be to perform CT imaging and use the features such as peripheral distribution, ground-glass opacity and vascular thickening of the CT images for diagnosis [Bai 2020].

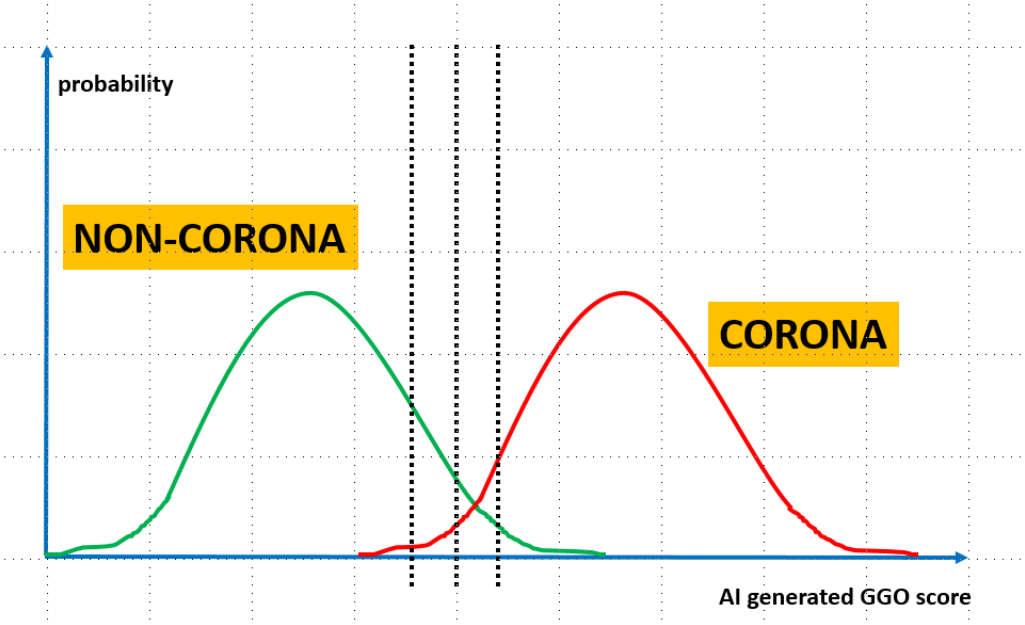
Assume the scientists designed an alternative AI system, which takes in a CT image, recognizes the ground-glass opacity (GGO) feature, and performs the diagnosis in a few seconds. However, there is a trade-off between efficiency and accuracy, so we have to evaluate how much we can trust the system.

(Simulated) Dataset: 100 patients were both tested by RT-PCR and the CT-based AI system: 51 patients were diagnosed by RT-PCR (the gold-standard) as positive (True) while 49 tested negative (False). The raw GGO values were collected from the AI system before making any thresholding.

The data is saved in data\GGO_value.mat and data\diagnosis.mat respectively.

Question 1 [3 points]

Assume the probability of positive and negative patients follow Gaussian distributions (see the two schematic plots below). Notice there is overlap between the two distributions (which means if we take different thresholds, we'll obtain different prediction results).



a) Using MATLAB, load the data and find the mean and standard deviation (std) of the Gaussian that models the positive distribution for 51 subjects.
b) Find the mean and std of the Gaussian that models the negative distribution for 49 subjects by MATLAB.
c) Show the plot of the two distributions in MATLAB (using the mean and std values found in parts a and b). Label your axes to obtain a figure similar to schematic plot above. Hint: use MATLAB's **normpdf** function.

Question 1. Your Answers:

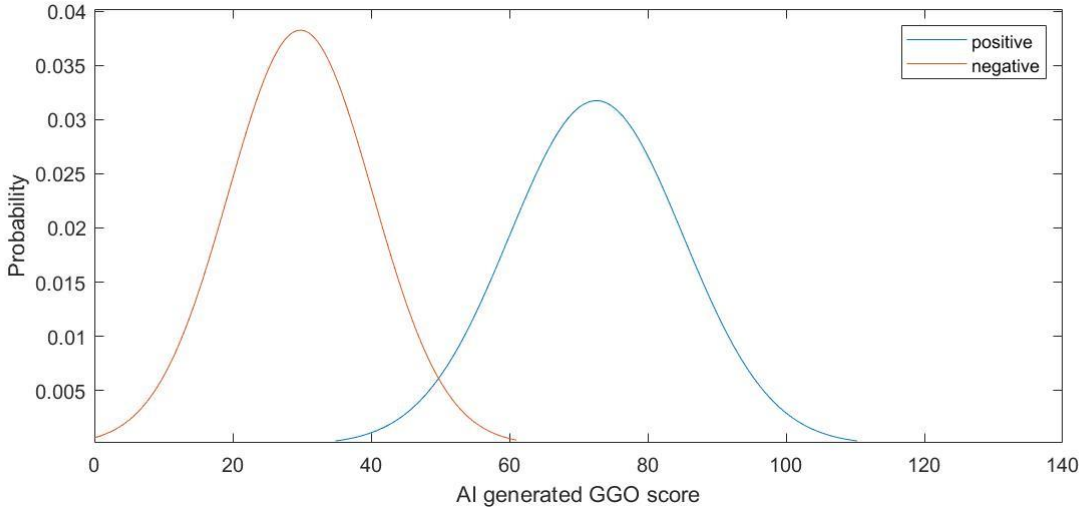
a) Mean of positive subjects: 72.5686

   Std of positive subjects:12.5686

b) Mean of negative subjects: 29.7959

   Std of negative subjects: 10.4323

c) Paste plot here:



Paste Code Here:
Loading data:

```
load(['data\', 'diagnosis.mat']);
load(['data\', 'GGO_value.mat']);

Calculating mean and std:
positive = zeros(1, 51);
negative = zeros(1, 49);
pos = 0;
neg = 0;

for i = 1:100
   if diagnosis(i) == 1
      pos = pos +1;
      positive(pos) = GGO_values(i);
   end

   if diagnosis(i) == 0
      neg = neg+1;
      negative(neg) = GGO_values(i);
   end
end

mean_pos = mean(positive);
std_pos = std(positive);

mean_neg = mean(negative);
std_neg = std(negative);

display(['Mean of positive subjects: ', num2str(mean_pos)]);
display(['Std of positive subjects: ', num2str(std_pos)]);
```

```
display(['Mean of negative subjects: ', num2str(mean_neg)]);
display(['Std of negative subjects: ', num2str(std_neg)]);



Choose the threshold range and step
Build the distribution function
x_pos = [mean_pos-4*std_pos : 0.1 : mean_pos+4*std_pos];
x_neg = [mean_neg-4*std_neg : 0.1 : mean_neg+4*std_neg];

y_pos = normpdf(x_pos, mean_pos, std_pos);
y_neg = normpdf(x_neg, mean_neg, std_neg);

Plot your figures
plot (x_pos, y_pos);
xlabel('AI generated GGO score')
ylabel('Probability')
hold on
plot (x_neg, y_neg);
legend({'positive','negative'})
hold off
```

Question 2 [3 points]

Given the 2 Gaussian distributions in Question 1, the goal is to construct the corresponding ROC curve.

a) Choose your threshold values to construct the ROC curve. Make sure your choice contains at least 10 different values.
b) Plot the ROC curve with TP rate (TPR) in percentage along the vertical axis, vs. FPR along the horizontal, using both the erf table and the **normcdf** function.

Note:

- The ROC should show the operating points for equally-separated thresholds.
- Do not use the raw data to calculate the operating points, instead use the Gaussian distributions.
- To calculate needed integrals (i.e. CDF), refer to the lecture slides and make use of the values given in the provided file: **erf_tables.pdf**. Note: for $x < 0$, erf is negative and equal to -erf(-x) as read from the table.
- Use MATLAB to plot and make sure that the operating points are clearly visible. Double check your answers using MATLAB's built-in function **normcdf** to calculate the integrals over a Gaussian distribution. You can use a finer threshold grid so the ROC curve will look smoother.

Question 2. Your Answers:

a) Choose threshold Values of the ROC Operating points:

(0 : 1 : 110)

b) Explain how you used erf table for three example thresholds values:

x1 = 1-TP = normcdf(20, 72.5686, 12.5686) = 0.5 + 0.5*erf((20-72.5686)/(12.5686*√2)

= 0.5 + 0.5*erf(-2.96) = 0.000015

y1 = TN = normcdf(20, 29.7959, 10.4323) = 0.5 + 0.5*erf((20-29.7959)/(10.4323*√2)

= 0.5 + 0.5*erf(-0.66) = 0.17531

x2 = 1-TP = normcdf(50, 72.5686, 12.5686) = 0.5 + 0.5*erf((50-72.5686)/(12.5686*√2)

= 0.5 + 0.5*erf(-1.27) = 0.036245

y2 = TN = normcdf(50, 29.7959, 10.4323) = 0.5 + 0.5*erf((50-29.7959)/(10.4323*√2)

= 0.5 + 0.5*erf(1.37) = 0.973655

x3 = 1-TP = normcdf(100, 72.5686, 12.5686) = 0.5 + 0.5*erf((100-72.5686)/(12.5686*√2)

= 0.5 + 0.5*erf(1.54) = 0.985295

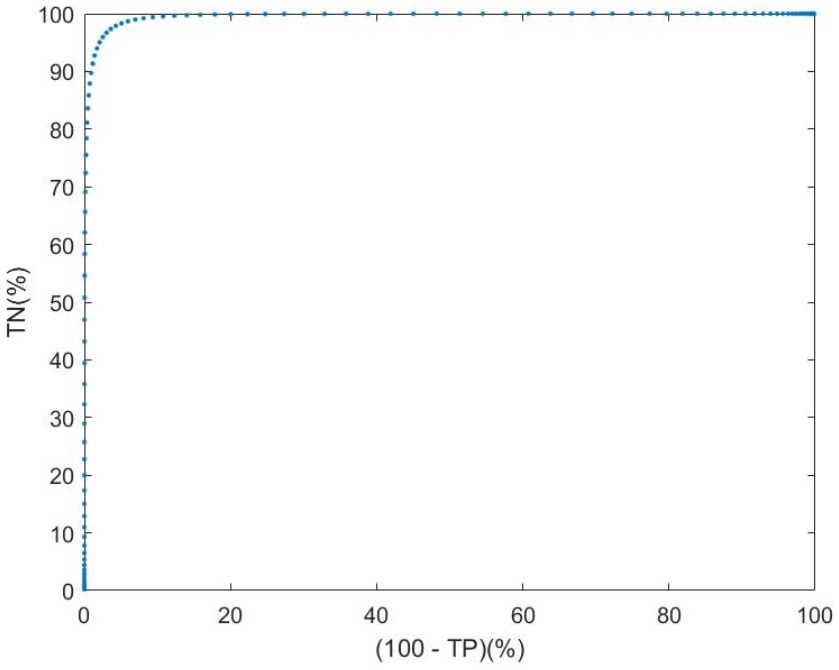
y3 = TN = normcdf(100, 29.7959, 10.4323) = 0.5 + 0.5*erf((100-29.7959)/(10.4323*√2)

= 0.5 + 0.5*erf(4.76) = 1

Paste MATLAB Code for plotting the ROC curve (use scattered points instead of line segments) Here:

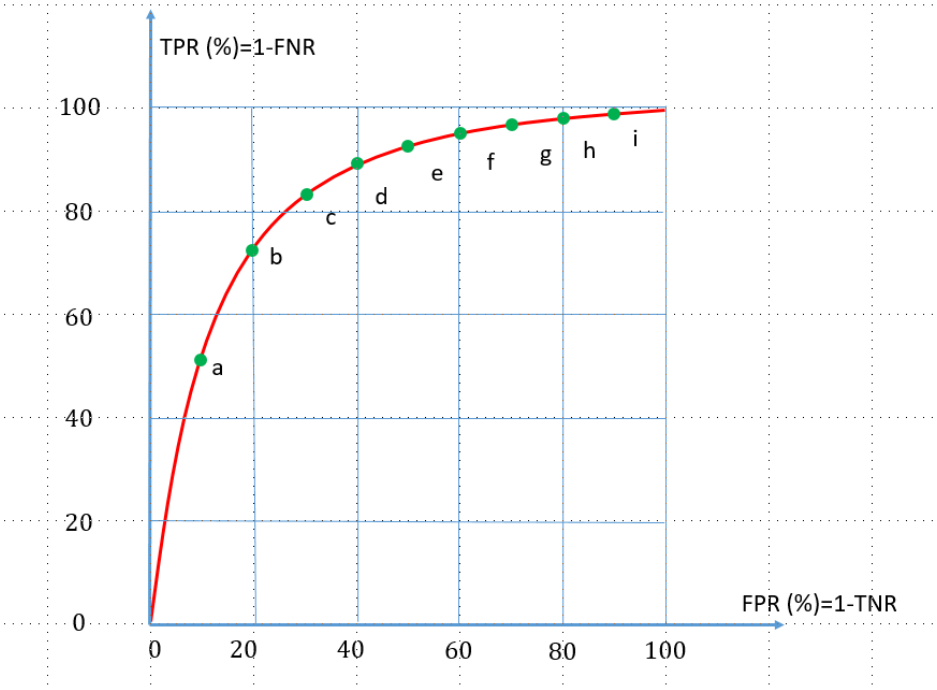
```
threshold = (0 : 1 : 110);

for i = 0:110
   threshold(2, i+1) = 100*normcdf(i, mean_neg, std_neg);
   threshold(3, i+1) = 100*normcdf(i, mean_pos, std_pos);
end

plot(threshold(3, :), threshold(2, :), '.')
xlabel('(100 - TP)(%)')
ylabel('TN(%)')
```

Paste ROC Figure here:



Question 3 [4 points]

Now we look at a dataset collected by a deployed beta-version of the AI system, which resulted in the following ROC curve:



Your task is to control the mis-diagnosis ratio, by reaching a trade-off between TPR and FPR.

1. In particular, you need to tune certain hyper-parameters (e.g. threshold T) for the decision system so that such that: $FNR \leq 20\%$ with the lowest possible FPR. Among the 9 possible operating points (green dots) in the figure above, which one would you choose to satisfy the requirement? Please explain.

2. Assume 20 AI-diagnosed patients had OGG values:
   $V = [70, 60, 30, 80, 40, 20, 50, 90, 85, 45, 75, 65, 55, 15, 35, 45, 45, 65, 65, 75]$.
   Then these 20 patients underwent the more reliable RT-PCR test, which returned, 72 hours later, the following diagnoses, which we regard as "truth":
   C = [P, P, N, P, N, N, N, P, P, N, P, P, P, N, N, N, N, P, P, N].
   where (P: positive, i.e. COVID19; N: negative, i.e. non-COVID19)

   Choose the threshold so that the AI-classification results would have $FNR \leq 10\%$ and $FPR \leq 40\%$? Justify your choice. Note: Calculate FNR using the data points and not a fitted, Gaussian or other, distribution.

3. Using the threshold, you chose in question 2. Answer the following questions:
   a. How many were misdiagnosed?
   b. How many sick patients were diagnosed as healthy?
   c. How many healthy patients were diagnosed as sick?
   d. What's the false negative ratio?

4. Calculate entries of 2x2 confusion matrix for the 10 patients, use the number of patient in the entries, e.g. number of patients that are N but were misdiagnosed as P, etc.

5. Now draw another confusion matrix and enter the percentages instead, i.e. out of 100% negative cases, what percent were correctly classified as N, etc.

Question 3. <u>Your Answers:</u>

1. Choose operating point
$FNR \leq 20\%, TPR = 1 - FNR \geq 80\%$, with the lowest possible FPR, I would choose c.

2.
   a. List and sort the positive and negative OOG values (you can use MATLAB command sort here)

15	20	30	35	40	45	45	45	50	75
N	N	N	N	N	N	N	N	N	N
55	60	65	65	65	70	75	80	85	90
P	P	P	P	P	P	P	P	P	P

   b. How to choose a threshold so that FNR<=10%?
   Choose any number greater than 50, FNR >= 10%.

   c. How to choose a threshold so that FPR<=40%?
   Choose any number less than 65, FPR <= 20%.

   d. What's your choice of the final threshold?
   Any number between 50 and 65, such as 52.

3.
   a. Misdiagnosed number: 1

   b. Sick diagnosed as healthy: 0

   c. Healthy diagnosed as sick: 1

   d. FNR = 10%

4. Confusion matrix in number

		Decision Model	
		Test Positive	Test Negative
Truth	Condition Positive	10	1
	Condition Negative	0	9

5. Confusion matrix in percentage

		Decision Model	
		Test Positive	Test Negative
Truth	Condition Positive	100%	10%
	Condition Negative	0%	90%

References

[Bai 2020] H. X. Bai et al., “Performance of radiologists in differentiating COVID-19 from viral pneumonia on chest CT”, Radiology, 2020. DOI: https://doi.org/10.1148/radiol.2020200823