DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Homework Assignment No. 07:

HW No. 07: Information Theory and Statistical Significance

submitted to:

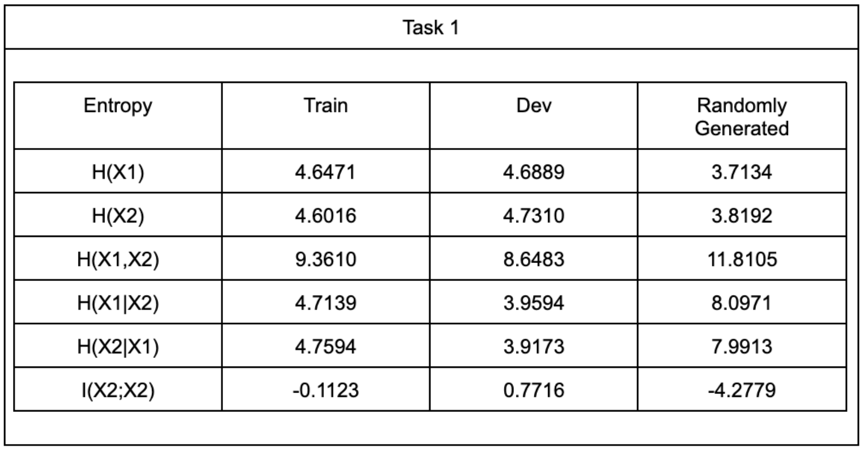
Professor Joseph Picone
ECE 8527: Introduction to Pattern Recognition and Machine Learning
Temple University
College of Engineering
1947 North 12th Street
Philadelphia, Pennsylvania 19122

March 8th, 2022

prepared by:

Gavin Koma
Email: gavintkoma@temple.edu

A. TASK 1



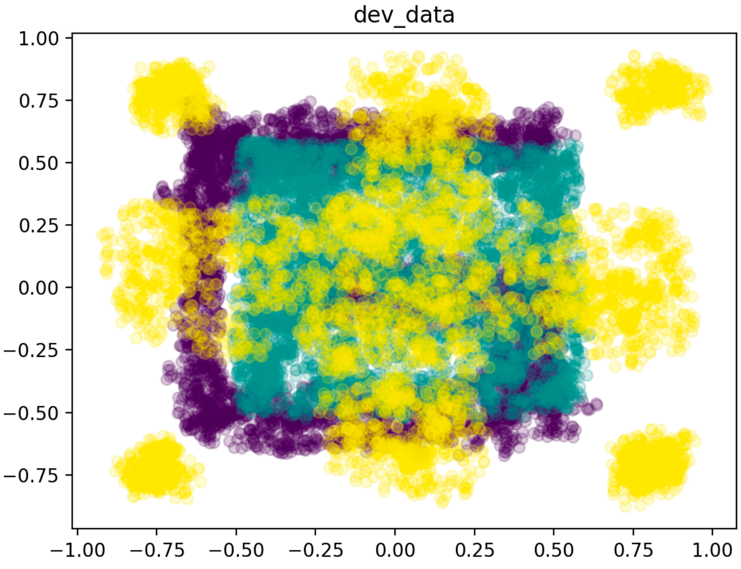
Task 1

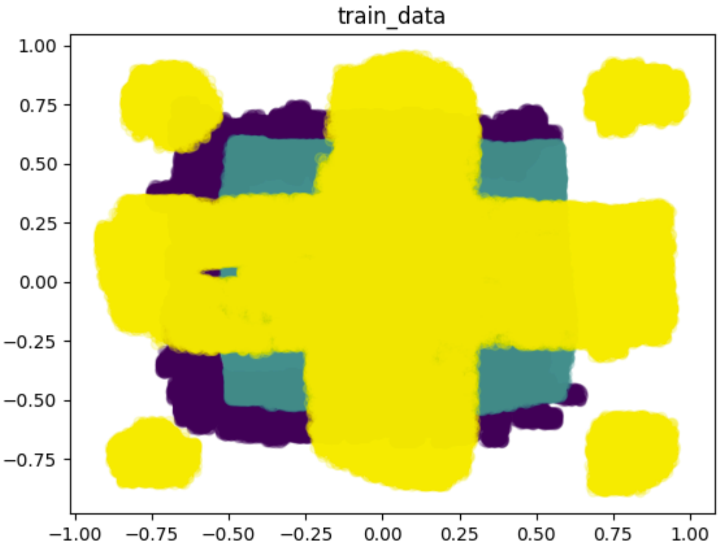
Entropy	Train	Dev	Randomly Generated
H(X1)	4.6471	4.6889	3.7134
H(X2)	4.6016	4.7310	3.8192
H(X1,X2)	9.3610	8.6483	11.8105
H(X1|X2)	4.7139	3.9594	8.0971
H(X2|X1)	4.7594	3.9173	7.9913
I(X2;X2)	-0.1123	0.7716	-4.2779

The goal of Task1 is to assess a two-dimensional dataset with feature vectors: $[x_1, x_2]$; and then spend time to quantize each element of each vector to a set of 128 discrete values. Then, we are to explore the entropy between varying vectors. We are given the equation do this already and is as such:

$$y = round(\left(\frac{x}{x_{max} - x_{min}}\right) * 128)$$

As with previous assignments, I first began by plotting the original data for visualization purposes. The graphs are plotted in a simple manner and so code has been excluded but the plots have been attached below:

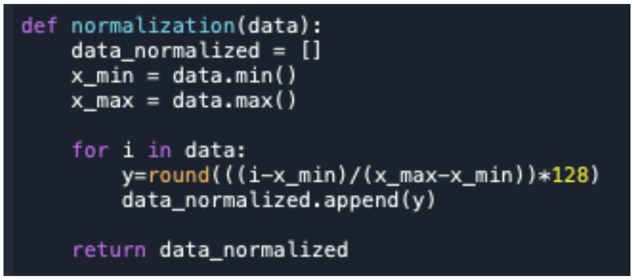




We first need to normalize our data using the formula that was provided in the assignment. Doing so will allow us to scale each of the given datapoints to the range of its corresponding vector. Ergo, our range of values will be quantized to 0-128 instead of the current range. One important thing to note is how we will explore the entropy of the vectors. Here, we will dive further into Shannon entropy; with a quick google search, Shannon entropy may be defined as a measure of uncertainty of occurrence of a certain event when given partial information regarding the system. This idea can be mathematically presented as:

$$H(x) = -\sum p(x) * \log(p(x))$$

Luckily, we do not need to define our own function to apply this concept and instead we can employ the built-in python entropy function within the scipy library. A simple function was written in order to normalize the data. This facilitated the process by allowing me to just pass any data to this function and return with a dataframe of normalized values. The code to do this is attached below:

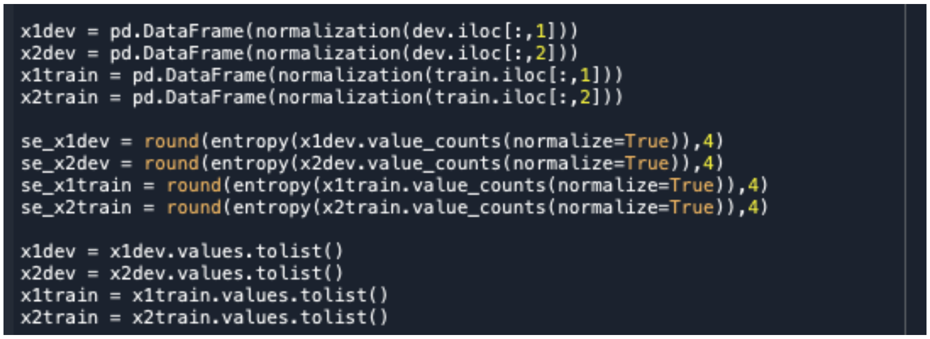


```
def normalization(data):
    data_normalized = []
    x_min = data.min()
    x_max = data.max()

    for i in data:
        y=round(((i-x_min)/(x_max-x_min))*128)
        data_normalized.append(y)

    return data_normalized
```

Once all needed data has been passed through our normalization function, we can continue to organize the data and calculate the entropy of the vectors as follows:

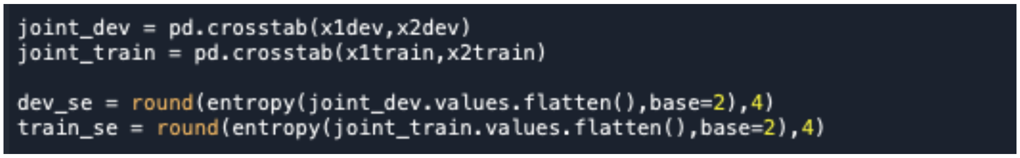


```
x1dev = pd.DataFrame(normalization(dev.iloc[:,1]))
x2dev = pd.DataFrame(normalization(dev.iloc[:,2]))
x1train = pd.DataFrame(normalization(train.iloc[:,1]))
x2train = pd.DataFrame(normalization(train.iloc[:,2]))

se_x1dev = round(entropy(x1dev.value_counts(normalize=True)),4)
se_x2dev = round(entropy(x2dev.value_counts(normalize=True)),4)
se_x1train = round(entropy(x1train.value_counts(normalize=True)),4)
se_x2train = round(entropy(x2train.value_counts(normalize=True)),4)

x1dev = x1dev.values.tolist()
x2dev = x2dev.values.tolist()
x1train = x1train.values.tolist()
x2train = x2train.values.tolist()
```

In order for us to calculate the joint probability, we needed to generate a joint probability function made of the two feature vectors:



```
joint_dev = pd.crosstab(x1dev,x2dev)
joint_train = pd.crosstab(x1train,x2train)

dev_se = round(entropy(joint_dev.values.flatten(),base=2),4)
train_se = round(entropy(joint_train.values.flatten(),base=2),4)
```

It is possible to deduce the remaining joint entropies by rearranging our equation to determine them. Consider that we know H(X1, X2) and that we also now know H(X1) and can also obtain H(X1|X2). We can obtain the conditional entropy of X2 given X1 by performing the following:

$$H(X2|X1) = H(X1, X2) - H(X1)$$

Finally, we calculate I(X1;X2) as follows:

$$I(X1; X2) = H(X1, X2) - H(X2|X1) - H(X1|X2)$$

Unfortunately, as seen in my table above, there are two negative values seen. This means that at least one of my entropies were calculated incorrectly. I was unable to determine the exact bug in my code after attempting to solve this problem and I have instead elected to acknowledge the problem but not submit my assignment late.

B. TASK 2

Task 2 involves assessing two systems (a baseline system and a new system) which assess performance of a dataset of 1,000 files. We are initially told that the baseline system gives us an error rate of 20.0% and the new system delivers and error rate of 19.0%. Our first goal is to determine if the new system, which delivers an error rate of 19.0%, is statistically significant at a confidence level of 80%.

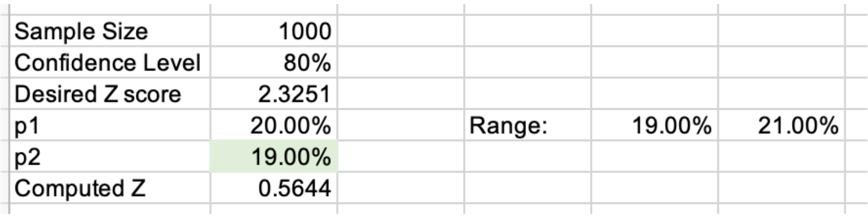
We can do this by assuming that P1 and P2 are 20.0% and 19.0%, respectively. It is also important to note that the value of N will be the same: 1,000 total datafiles. From here, we can utilize the Z-score formula which is as follows:

$$Z = \frac{(p_1 - p_2)}{\sqrt{(\frac{p_1(1-p_1)}{N_1} + \frac{p_2(1-p_2)}{N_2})}}$$

$$Z = \frac{(0.20 - 0.19)}{\sqrt{(\frac{0.20\,(1-0.20)}{1000} + \frac{0.19(1-0.19)}{1000})}}$$

$$Z = 0.56442$$

We are also provided with an Excel sheet that is capable of calculating our Z-score given a value for P1 and P2. We are able to see that the calculated computed Z score from this Excel sheet is the same as the Z score calculated from the above equation. Attached below is a screenshot of the Z-score.



Sample Size	1000				
Confidence Level	80%				
Desired Z score	2.3251				
p1	20.00%		Range:	19.00%	21.00%
p2	19.00%				
Computed Z	0.5644				

As discussed in class, we know that any positive scores that are in the z-table will correspond to some value which is less than the mean while the negative scores in the z-table will correspond to any values that are less than the mean. Recall that our calculated z-score is 0.5644 and we can utilize the following formula:

$$area = \frac{(1-c)}{2}$$

$$area = \frac{(1-0.80)}{2} = 0.1$$

This calculation allows us to find the z-score according to a corresponding area. In this case, the z-score for area (0.1) and the corresponding desired score for this area is 1.2820 which is larger than the computed z-score found above (0.5644). When we compare these two z-scores, we can make the decision for rejecting our null hypothesis. Let us state the following null and alternative hypothesis:

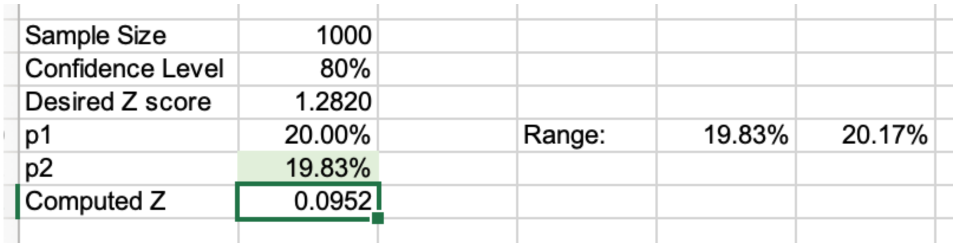
$$H_0 = \textit{There is no difference between the two systems.}$$

$$H_A = \textit{There is some statistical significance between the two systems.}$$

For part 1 of this Task, we can compare the two z-scores (1.2820 and 0.5644) and we can say that the z-score calculated via formula is larger than the z-score calculated by the area formula. We can state that P2 (the new system error rate) is significantly lower than P1.

The next portion of this assignment asks us to assess what the minimum decrease in error rate that will result in a statistically significant result. To make things easy, we can assess the various decreases in error rate that may result in something statistically significant.

To do this, we can utilize the Excel calculator provided and input varying numbers until we see a result that is *not* statistically significant. We are able to determine that a decrease in error rate of 00.17% will result in a not statistically significant result. Such a decrease will result in a z-score of 0.0952 which is less than our area z-score calculation and we are able to determine that P2 (the new system error) is not statistically significant when compared to P1 (the old system error)



Sample Size	1000				
Confidence Level	80%				
Desired Z score	1.2820				
p1	20.00%		Range:	19.83%	20.17%
p2	19.83%				
Computed Z	0.0952				

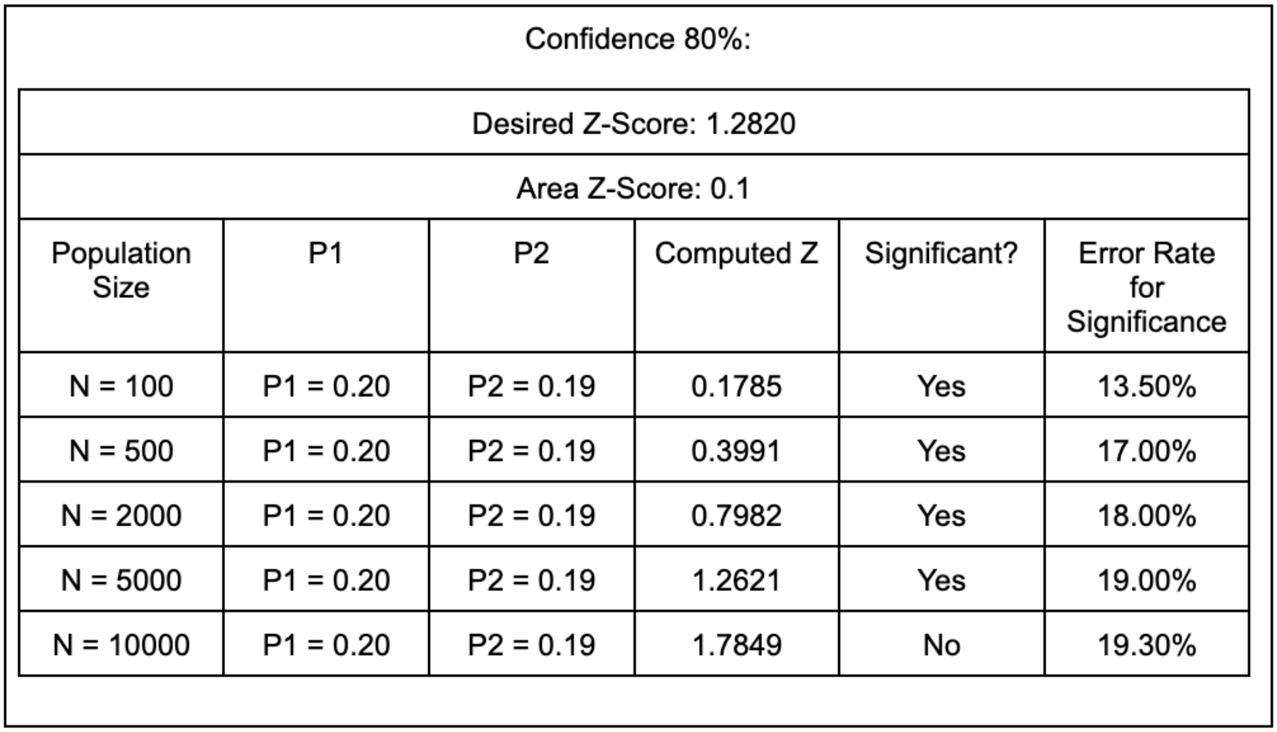
Our final goal of this task is to repeat the first two tasks for N = 100, 500, 2000, 5000, and 10000 with a confidence level of 80%, 85%, 90%, and 95%. Let us first refresh our memories as to what our chosen hypotheses are:

$$H_0 = \textit{There is no difference between the two systems.}$$

$$H_A = \textit{There is some statistical significance between the two systems.}$$

For ease of use, I wrote a simple python function that will calculate the score of both area z and computational z. This is just so I don't have to continue clicking through Excel but instead can just alter the variables. The equations are the same as utilized in the Excel sheet, so I have elected to not include a screenshot of the code.

Again, we will want to compare the area score to the other score. The scores are compiled below.

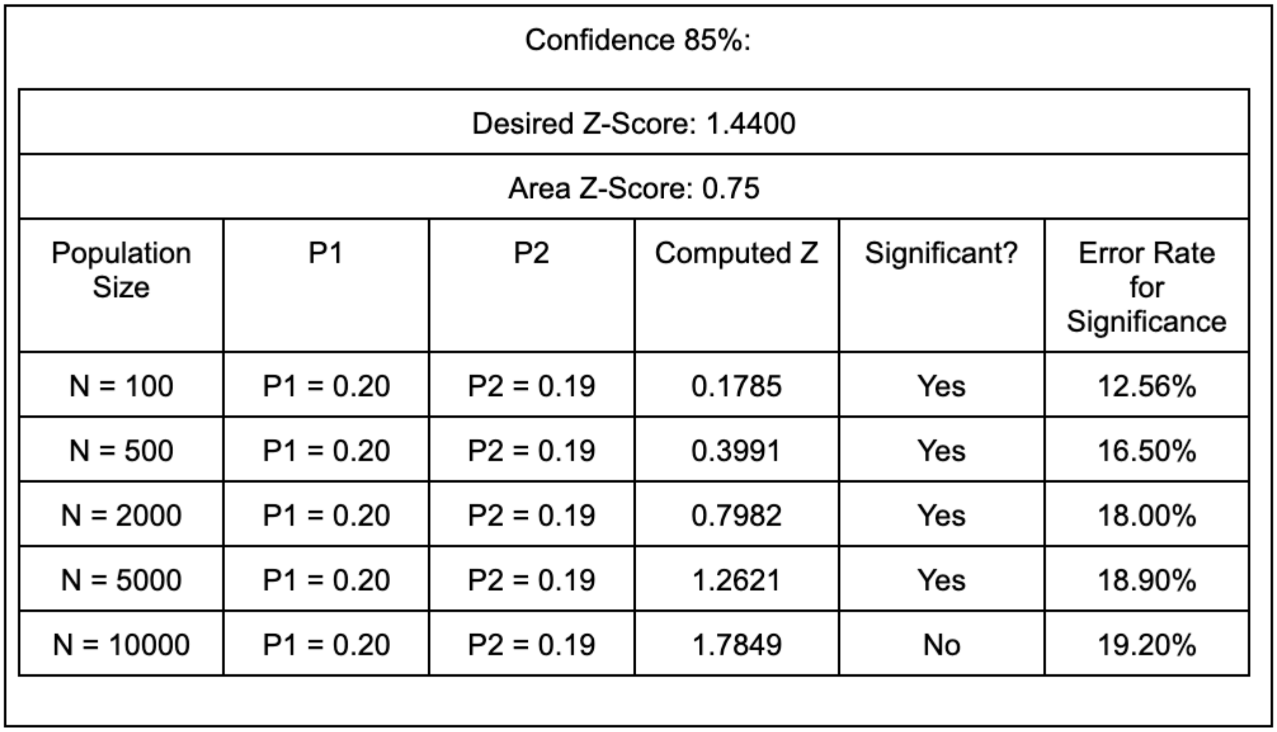


Confidence 80%:

Desired Z-Score: 1.2820

Area Z-Score: 0.1

Population Size	P1	P2	Computed Z	Significant?	Error Rate for Significance
N = 100	P1 = 0.20	P2 = 0.19	0.1785	Yes	13.50%
N = 500	P1 = 0.20	P2 = 0.19	0.3991	Yes	17.00%
N = 2000	P1 = 0.20	P2 = 0.19	0.7982	Yes	18.00%
N = 5000	P1 = 0.20	P2 = 0.19	1.2621	Yes	19.00%
N = 10000	P1 = 0.20	P2 = 0.19	1.7849	No	19.30%

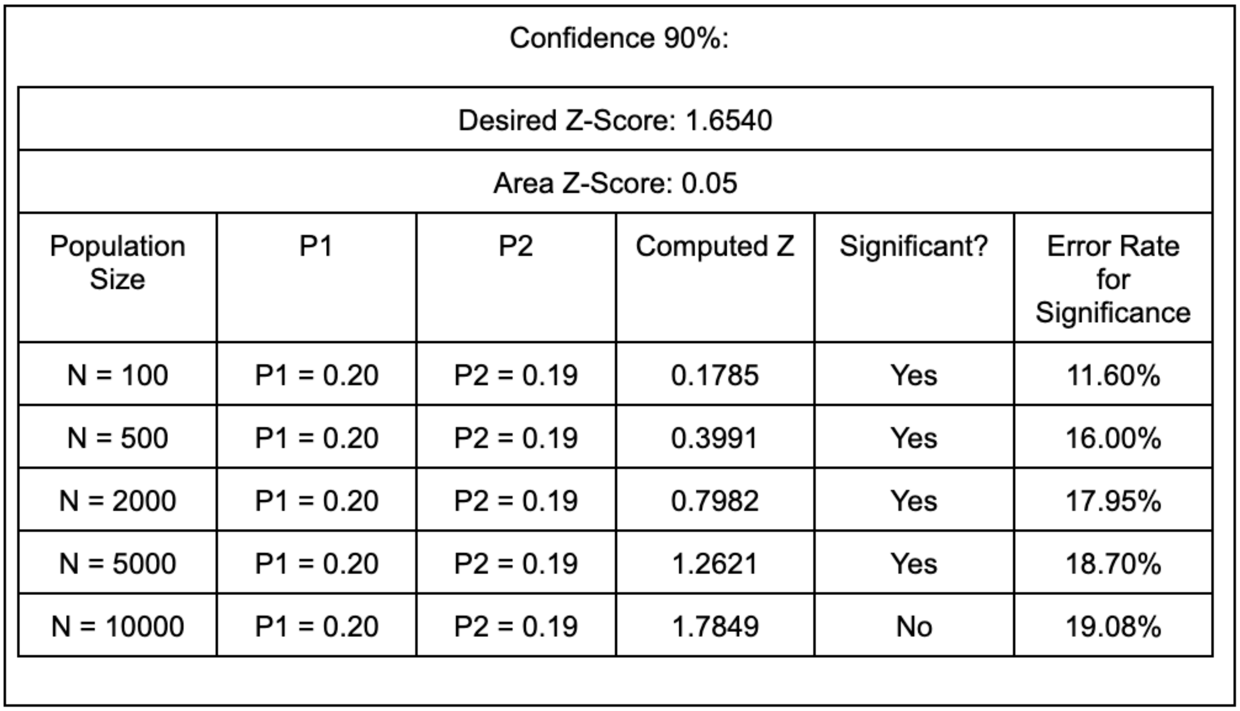


Confidence 85%:

Desired Z-Score: 1.4400

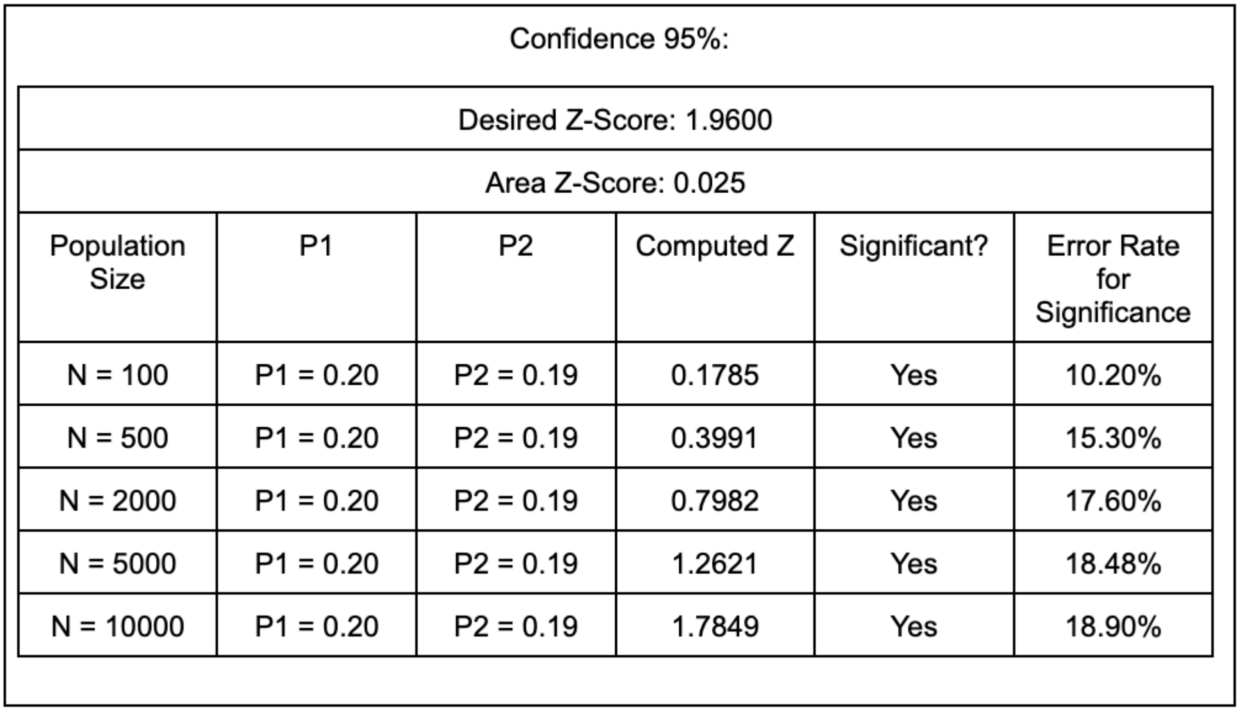
Area Z-Score: 0.75

Population Size	P1	P2	Computed Z	Significant?	Error Rate for Significance
N = 100	P1 = 0.20	P2 = 0.19	0.1785	Yes	12.56%
N = 500	P1 = 0.20	P2 = 0.19	0.3991	Yes	16.50%
N = 2000	P1 = 0.20	P2 = 0.19	0.7982	Yes	18.00%
N = 5000	P1 = 0.20	P2 = 0.19	1.2621	Yes	18.90%
N = 10000	P1 = 0.20	P2 = 0.19	1.7849	No	19.20%



Confidence 90%:

Desired Z-Score: 1.6540					
Area Z-Score: 0.05					
Population Size	P1	P2	Computed Z	Significant?	Error Rate for Significance
N = 100	P1 = 0.20	P2 = 0.19	0.1785	Yes	11.60%
N = 500	P1 = 0.20	P2 = 0.19	0.3991	Yes	16.00%
N = 2000	P1 = 0.20	P2 = 0.19	0.7982	Yes	17.95%
N = 5000	P1 = 0.20	P2 = 0.19	1.2621	Yes	18.70%
N = 10000	P1 = 0.20	P2 = 0.19	1.7849	No	19.08%



Confidence 95%:

Desired Z-Score: 1.9600					
Area Z-Score: 0.025					
Population Size	P1	P2	Computed Z	Significant?	Error Rate for Significance
N = 100	P1 = 0.20	P2 = 0.19	0.1785	Yes	10.20%
N = 500	P1 = 0.20	P2 = 0.19	0.3991	Yes	15.30%
N = 2000	P1 = 0.20	P2 = 0.19	0.7982	Yes	17.60%
N = 5000	P1 = 0.20	P2 = 0.19	1.2621	Yes	18.48%
N = 10000	P1 = 0.20	P2 = 0.19	1.7849	Yes	18.90%

We see that the minimum rate of decreases as we increase the confidence interval. Given lower confidence intervals, we are giving room for more error in our system while a smaller confidence interval decreases the room for error. We see this trend through all of the included charts above and it is an important thing to note for this task. We are able to determine significance by comparing the computed z-score to the area z-score that can be found in a z-table. We are able to reject our null hypothesis that there is no difference between the two systems, and we are able to state that there is some difference given a certain error rate. To do so, if our computed z-score is *less* than the desired z-score (found in the table) then we can reject our null hypothesis.

C. CONCLUSION

The goal of this assignment was to gain a better understanding of information theory and statistical significance. I particularly enjoyed the second part of this assignment, but I had spent many hours struggling with the first portion of this assignment. As previously mentioned, two of my entropy values are negative, but I have been unable to find the bug in my code that had resulted in this. I also had to wait a notably long amount of time for the entropy calculation to output from the training set due to the sheer size of it. We do see relatively low values for entropy in our vector columns in Task 1 which is understandable. These vectors have shape to them. It is strange that the randomly generated ones show lower though. I would have initially thought that this would be generally higher because of the randomness found within their generation.

I think this assignment helped to reinforce the importance of statistical significance as well as the role of entropy in datasets, however, I do think that I will have to do more reading to gain a full understanding of this topic. I worry that my inability to locate the bug in my code as well as my initial confusion with the entropy values will inhibit my future work with tasks like these.