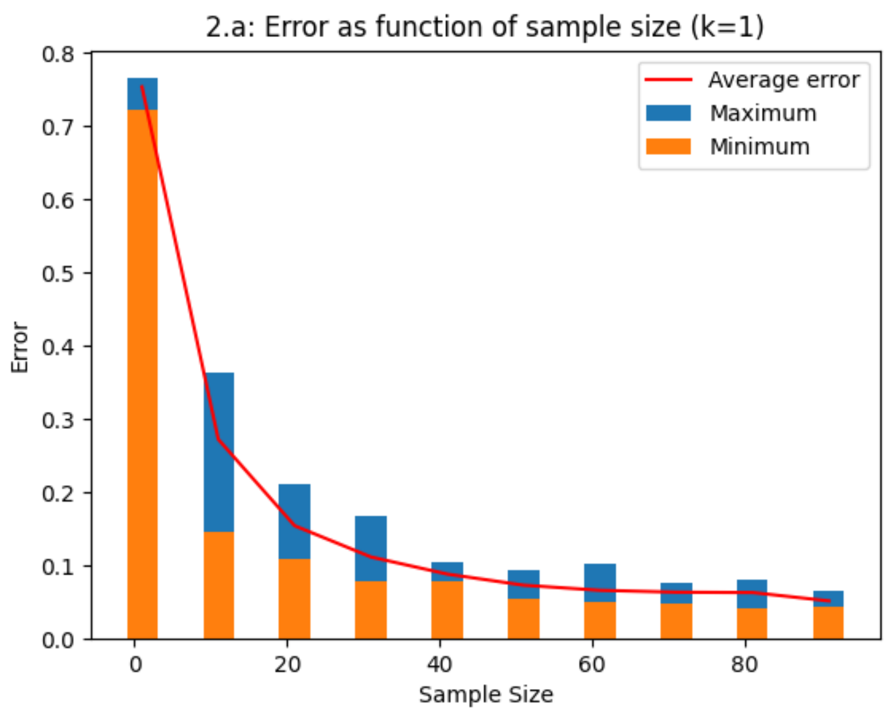
Question 1:
(code attached in zip file)

Question 2:
a. (*code attached in zip file - 'def ex_2a'*)



b. Yes. We can see that there is a decrease in the average error as the sample size grows. The reason for that decrease is yielded by the fact that k=1 which means our KNN algorithm is actually an NN algorithm. We also know that NN is an ERM algorithm and therefore:

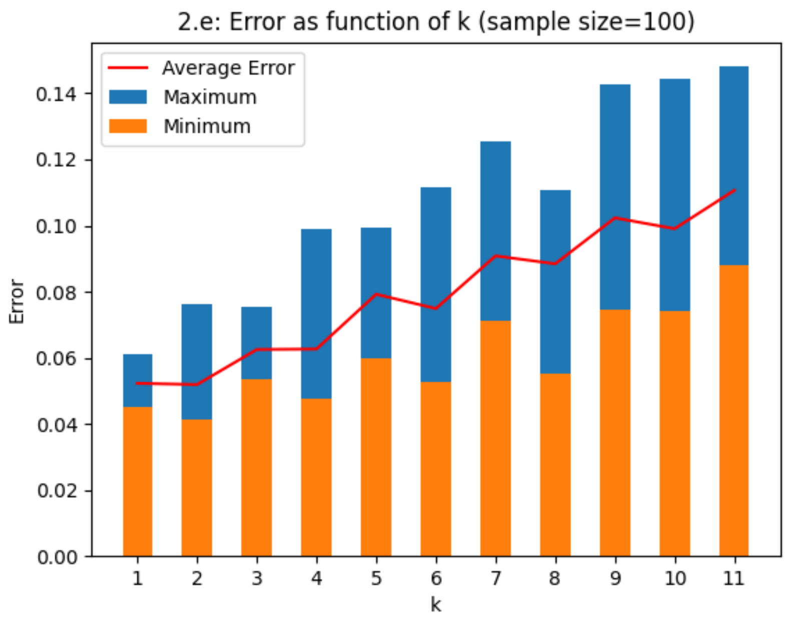
$$err(h,S) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}[h(x_i) \neq y_i]$$

Thus, as m grows, the expression for the error decreases.

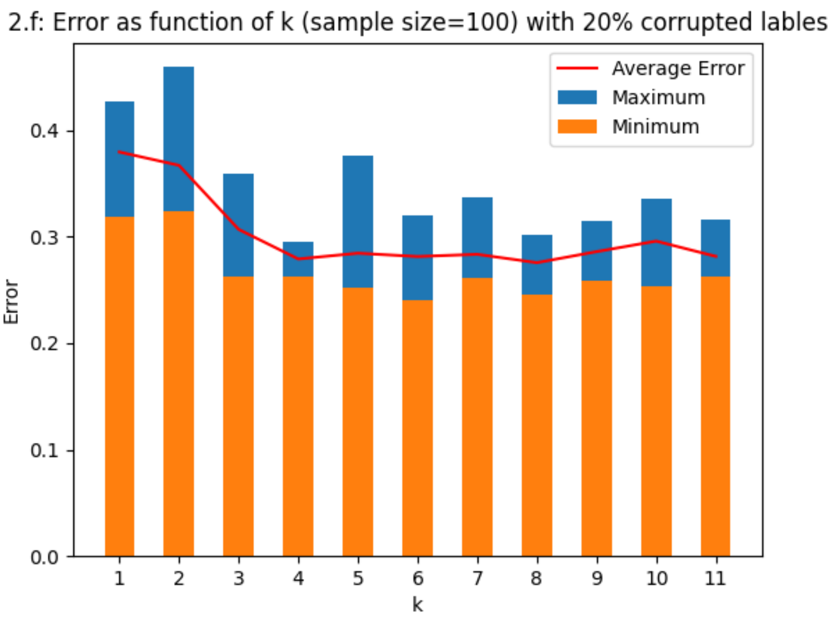
c. Yes. Since every time we use a sample size made of different random training examples, we add a measure of randomness to our run. Some sets of samples might "represent" our distribution better than others.

d. Yes, we can see a clear change in the size of the bars and we also see that the trend is similar to the trend of the average error. This decrease is an outcome of the increase in the sample size. What happens is that as we add more examples to our training set, the likelihood for having a NN of the same label increases as well. The reason is that in the metric we use(Euclidean distance in $\mathbb{R}^{784}$), pictures of the same digit tend to be closer to each other than to others. Thus, when having a test example of some digit, it is likely to be 'surrounded' by the same digits.

e.*(code attached in zip file - ‘def ex_2e')*



f.*(code attached in zip file - ‘def ex_2f')*



g. Yes. We can see that in section 2.e we have an increase in the error while in the other section we see a decrease for the error.
The reason is that, KNN is **not** an ERM algorithm. That means that KNN doesn’t try to minimize the error and we can clearly see it in the graph above (2.e).
When corrupting the labels(2f) randomly, we see that the error values are higher but there is a decrease as k grows. When using NN, if our test example is 'close' to some corrupted training example, it would be labeled as the same wrong label. However, in KNN, the corrupted labels have less impact on the algorithm because they are only 20% percent of all examples while we look at a whole group of examples(the neighbors).