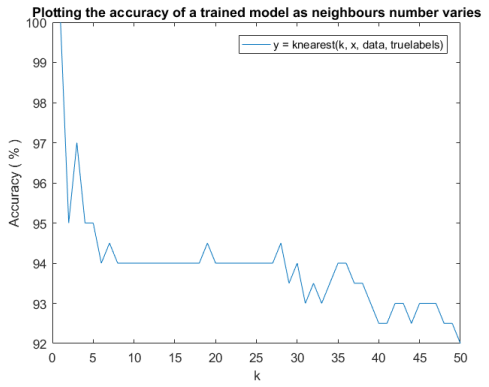
COMP24111

Exercise 2

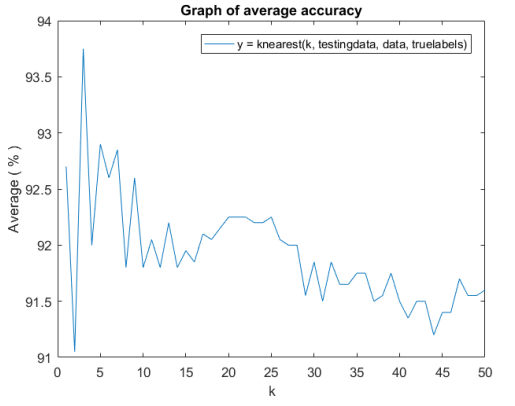
REPORT



Part 1.

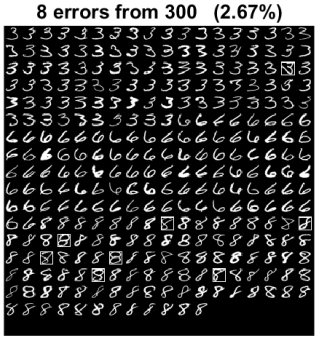
b) The best case is, as you can see in the graph, when k = 1 as the prediction has 100% accuracy. This is only training so it actually reaches 100%. And as I increased the number of neighbours to 50 you can see the accuracy of the prediction dropping.

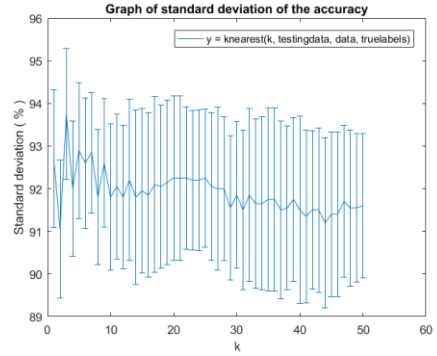
c) Comparing this to the first plot we can observe that this time we don't get 100% accuracy anymore because the train dataset is different from the test dataset. Also we get lower values throughout the whole plot even tho we have the same number of neighbours plotted.

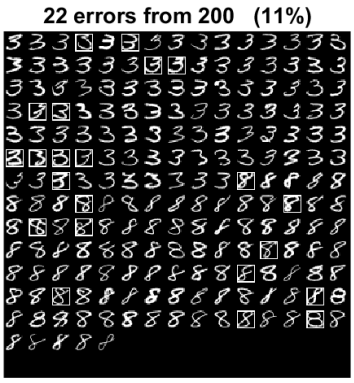


d) A decreasing trend in average accuracy can be found in the first graph as well. Therefore the same behavior applies here too but the data seems to be more fluctuant and less decreasing

e)We observe from the image that the easiest to classify is the number 6 followed by the number 3 and the number 8 is hardest with 7/8 errors. The confusion matrix tells us that 8's are always going to be harder to classify.



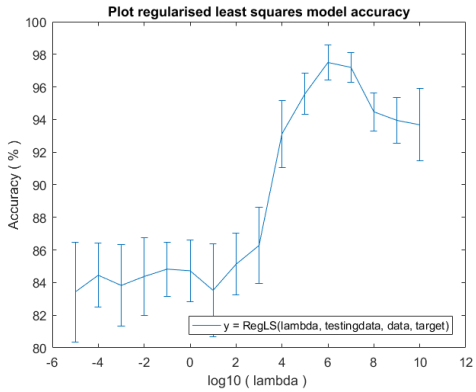




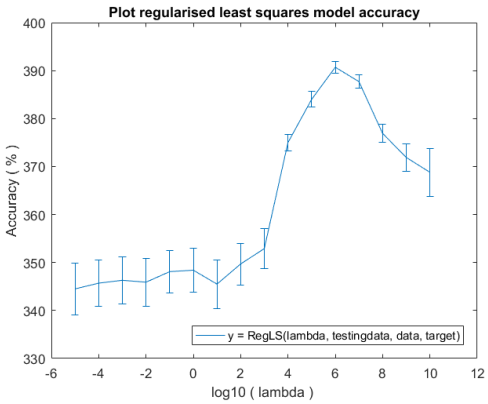
Part 2.

a) Compared to the k-NN approximation above there are more errors and it appears to have as make 8's correct as 3's so there is more balance in this method.

b)If we compare applying the models to the same data set multiple times we can see the same trend that the k-NN is more accurate and the regularized model although it is obviously more time consuming and executes far slower. On the other hand, again the later seems to find as many errors in 3's as in 8's and therefore seems to be more balanced.



c) We clearly identify a trend of lambda having a better output if it is higher than 10^4 and the best values seems to be around 10^6.



d) Randomizing the testing and training of the model we still get the same result meaning that the optimal value for lambda will be 10^6 .

e) Using the value for the hyperparameter found in the part d) we get error rates of even 1% so significantly increasing the accuracy .

Part 3

I used a 4-class with 1's , 3's, 6's, 8's. I had to design a target matrix that has 4 columns of 100 elements. It is designed to map the numbers by binary identifying each row (eg: 3's are mapped to be target(i ,:) = 1 0 0 0). I varied lambda again just like in part 2d to get the best value and as we can see it is again 10^6.