UUN: s1661552

Task 1

In this task the biggest problem I faced was the amount of time it was taking the algorithm to finish the classification. To tackle it, I used the tip provided in the coursework webpage to vectorize the calculations. I used $DI = (XX \quad \dots \quad XX) - 2 * X * Y^T + (YY \quad \dots \quad YY)^T$ tip, which helped to decrease the time it takes to calculate the distances from each training vector to each of the test vectors significantly. Unfortunately, the amount of RAM required for this operation was too large for my laptop and using 32-bit precision numbers didn't help. This could signify, that compromises could be made for the runtime to RAM usage ratio. On the other hand, so long as the programme ran on the DICE machines, I did not consider it to be a problem. To further minimise the runtime, I used as many built-in numpy functions as possible. Rather strangely, running it on DICE seems to sometimes cause the whole DICE account to crash, maybe due to the large amounts of RAM used or just flaws in DICE itself. As I've said, the memory consumption can be reduced by using for loops and, consequently, slowing the operation down. Running the whole classification for $Ks = [1, 3, 5, 10, 20]$ takes around 30 seconds on DICE. The best accuracy was achieved by using 3 closest neighbours. Using this number of closest neighbours resulted in the accuracy of 86.94%, while using 1 and 5 produced very similar results. As a side note, for markers convenience all the files required to run each classification system are included in each $Task*$ folder, even though some of them are the same across tasks. It would be more effective to split some of them to different packages in a real-life application.

Number of Ks	Time taken, s	Accuracy, %	N	N_{errs}
1	30.78	86.46	7800	1056
3	32.45	86.94	7800	1019
5	32.42	86.49	7800	1054
10	32.51	85.62	7800	1122
20	32.80	84.26	7800	1228