

## Inf2B-CW2

# Task 1 Report

### → my\_knn\_classify:

- to compute the squared euclidean distances between the training and test vectors, a fully vectorised approach was imperative for low computation time, as follows:
  - calculate (48000, 7800) matrix of distances between each training and test vector as specified in FAQ - this was crudely done in the first attempt - however, Sourav Dey<sup>1</sup> provides some insight into how to directly create this matrix in a much simpler way using full vectorisation techniques and a numpy trick for broadcasting the additions, per:

$$(x - y)^2 = x^2 + y^2 - 2xy$$

- to find the closest neighbours, utilise `argsort()`<sup>2</sup> to find their indices in the distances matrix, which will in turn correspond to the respective class in the Ctrn training vector class matrix
  - find the most common occurrence (mode) of class from training data cross-checked with k nearest neighbours, assign the outcome as the class prediction for the test vector

	distances	sort	total
Time elapsed approx (in seconds) [DICE environment, command line]	8.38	29.24	37.80

Statistics	k	N	Nerrs	acc
- from the data collected, the best accuracy is obtained when a value of $k = 3$ is chosen	1	7800	1083	86.10%
- there appear to be diminishing returns in accuracy when upwards of 5 nearest neighbours are selected, as more chaos is introduced into the system	3	7800	1049	86.54%
- works surprisingly well considering even just a single nearest neighbour	5	7800	1061	86.40%
	10	7800	1136	85.44%
	20	7800	1228	84.26%

<sup>1</sup> <https://medium.com/dataholiks-distillery/l2-distance-matrix-vectorization-trick-26aa3247ac6c>

<sup>2</sup> an attempt was made at foregoing full `argsort()` in order to shave a few seconds off the sorting time, namely by performing a partial sort using `argpartition()` to obtain the indices of the k lowest values in the array without actually sorting the whole array; however, these indices would need to be used to re-fetch values from the distance matrix, to then perform another sort, which unfortunately means that their original indices would have been lost in the process, rendering them unclassifiable; because of this, the idea of partitioning was not developed further

—