Inf2B-CW2

Task 2 Report

➔ my_bnb_classify:

- write results to new array based on threshold condition
- calculate probability of each class by dividing **class feature occurrences** by **total class occurrences**
- iterate through test vectors:
  - use naïve Bayes formula to calculate the $P(\vec{b}|C_k)$ as product $\prod_{i=0}^{D-1} P(b_i = 0|C_k)^{1-b_i} P(b_i = 1|C_k)^{b_i}$
  - this is done using `np.where()`, which takes a condition and returns the first argument in indices at which it is true, and the second in indices at which is is false - effectively recreating the terms $P(b_i = 0|C_k)^{1-b_i}$ and $P(b_i = 1|C_k)^{b_i}$ in `p0` and `p1`, respectively, as a `26x784` array of class feature probabilities for this vector
  - these are then collapsed via the `prod()` function to give a final `26x1` array of class probabilities
  - find the index at which the product is maximised and set that as prediction for class

---

Time elapsed approx (in seconds) [DICE environment, command line]	total
	0.99

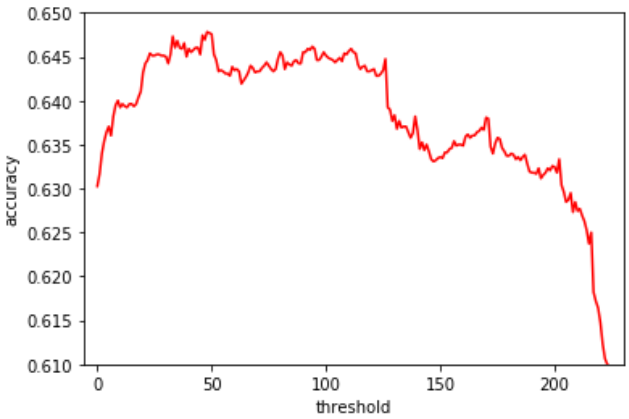
Statistics

(all possible thresholds were calculated and are displayed to the right)

- best accuracy within thresholds 35-55
- accuracy stays at 64% ± 1% between thresholds 0 to 205
- a “sharp” drop is observed at the threshold halfway point, 127

Comparisons

- considerably faster than K-NN
  - offers a quicker insight into data



(accuracy falls rapidly at threshold values greater than 230 [therefore omitted]; accuracy tends to ~3.8% [all vectors are classed as 0, akin to random chance] at threshold = 255)