

INF2B CW2 - Task 2

s1620208 - 13th April 2018

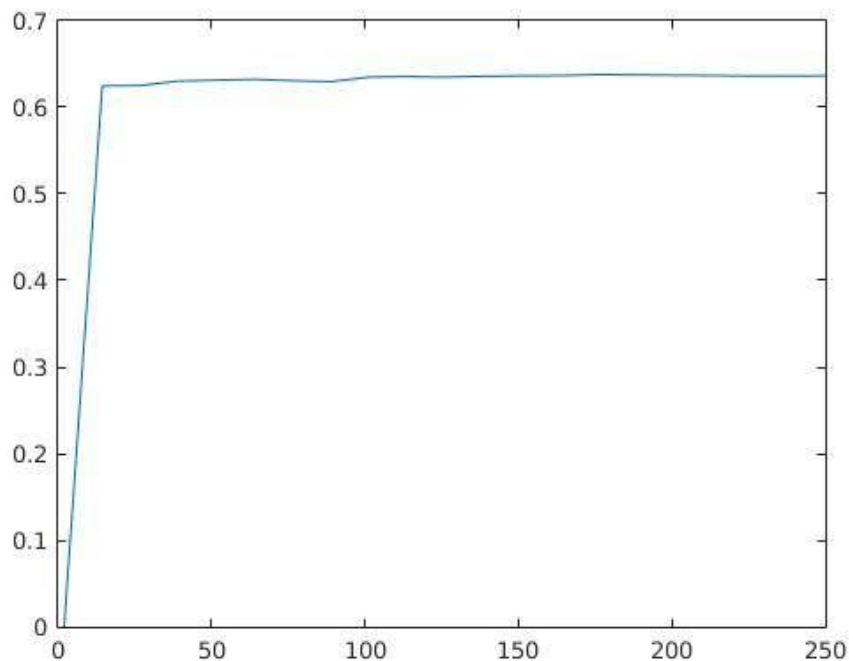
Optimisations within my code included calculating the inverse beforehand instead of per for loop. I also made use of vectorisation to mass perform calculations (such as the epsilon addition, or inverting) to make the whole operation run a lot faster.

```
for k in range(classes):
    # Remember Ctrn is a M-by-1 vector
    # row_indices is the list of row indices in Ctrn where we're dealing with this char
    row_indices, _ = np.where(Ctrn == k)

    probs[1][k] = Xtrn_binarised[row_indices].sum(axis=0) / len(row_indices)

# add smallest possible number to prevent issues with using 0
# this used to be in the above for loop too
probs[1][probs[1] == 0] = epsilon

# this used to be in the above for loop
probs[0] = 1 - probs[1]
```



My code for this runs within 8 seconds, making it difficult to plot a graph with a high resolution with the limited amount of remaining time. You can clearly see that a higher