Nabin Chapagain
1001551151 **Nov 9**

Training and testing on the pendigits dataset, with k=1.

classification accuracy=0.9743

Training and testing on the pendigits dataset, with k=3.

classification accuracy=0.9750

Training and testing on the pendigits dataset, with k=5.

classification accuracy=0.9763

Training and testing on the satellite dataset, with k=1.

classification accuracy=0.8935

Training and testing on the satellite dataset, with k=3.

classification accuracy=0.9042

Training and testing on the satellite dataset, with k=5.

classification accuracy=0.9045

Training and testing on the yeast dataset, with k=1.

classification accuracy=0.4959

Training and testing on the yeast dataset, with k=3.

classification accuracy=0.5179

Training and testing on the yeast dataset, with k=5.

classification accuracy=0.5514

Nabin Chapagain
1001551151 **Nov 9**

Task 1b

In this task, you are free to change any implementation options that you are not free to change in Task 1.

Changing k

I tried k = 7 with yeast_training and yeast_test files and the classification accuracy was 54.41% which was less than one with k = 5.

But, on using k = 9, the accuracy percentage went up to 55.51% which is the highest so far.

I changed k to 11 and now the accuracy went up to 57.20% **which shows a trend that accuracy percentage goes up if we include more nearest points in the process.**

Different distances

The result I have above is due to the use of Euclidean distance.

When I used hamming distance formula, the classification accuracy went down significantly to 34.06% which is huge setback to the accuracy obtained using Euclidean distance.

I then thought about using Mahalanobis Distance but I could not figure what covariance matrix to use for this situation.