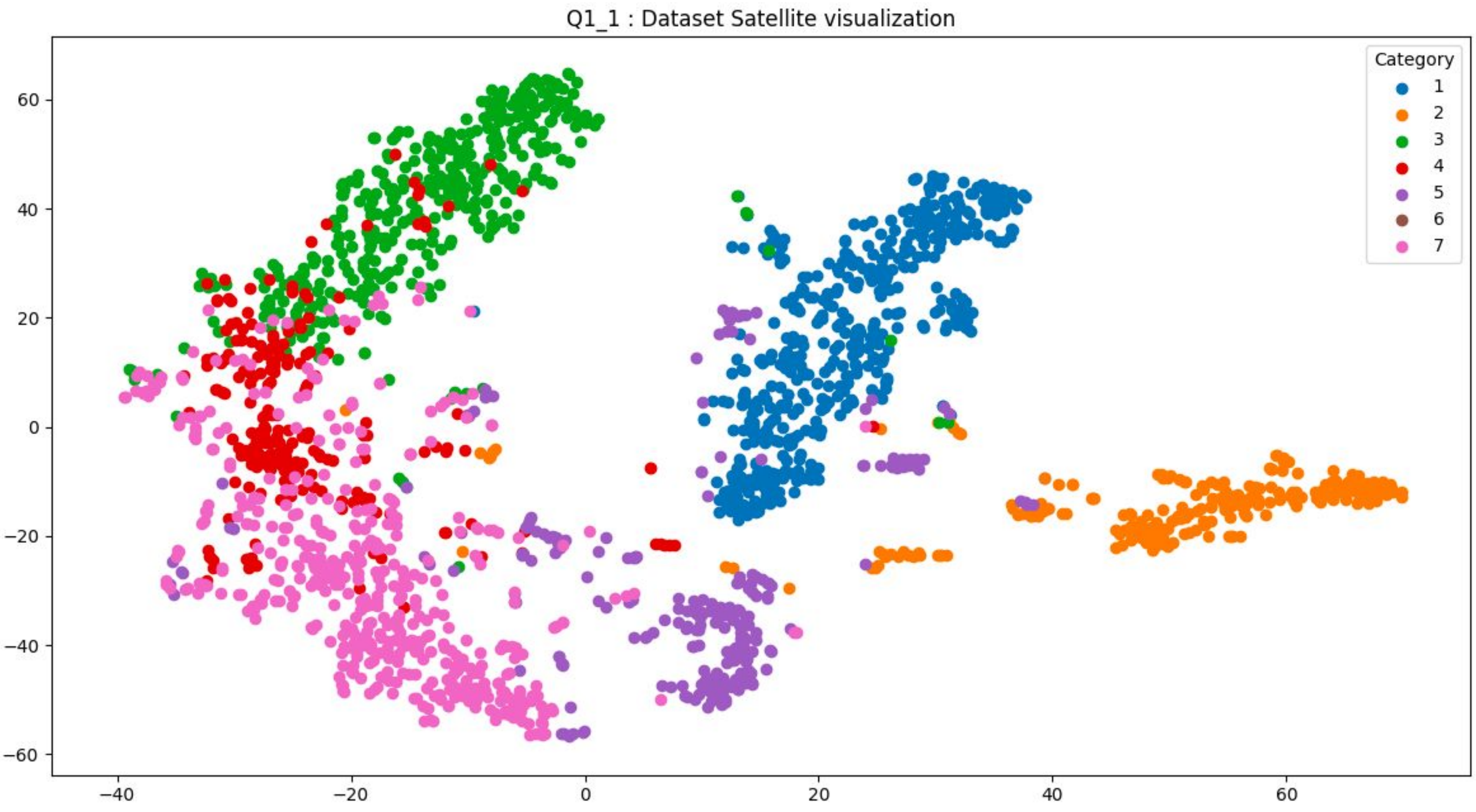
ML Assignment 5

Group 88 : Harkishan Singh (2017233)

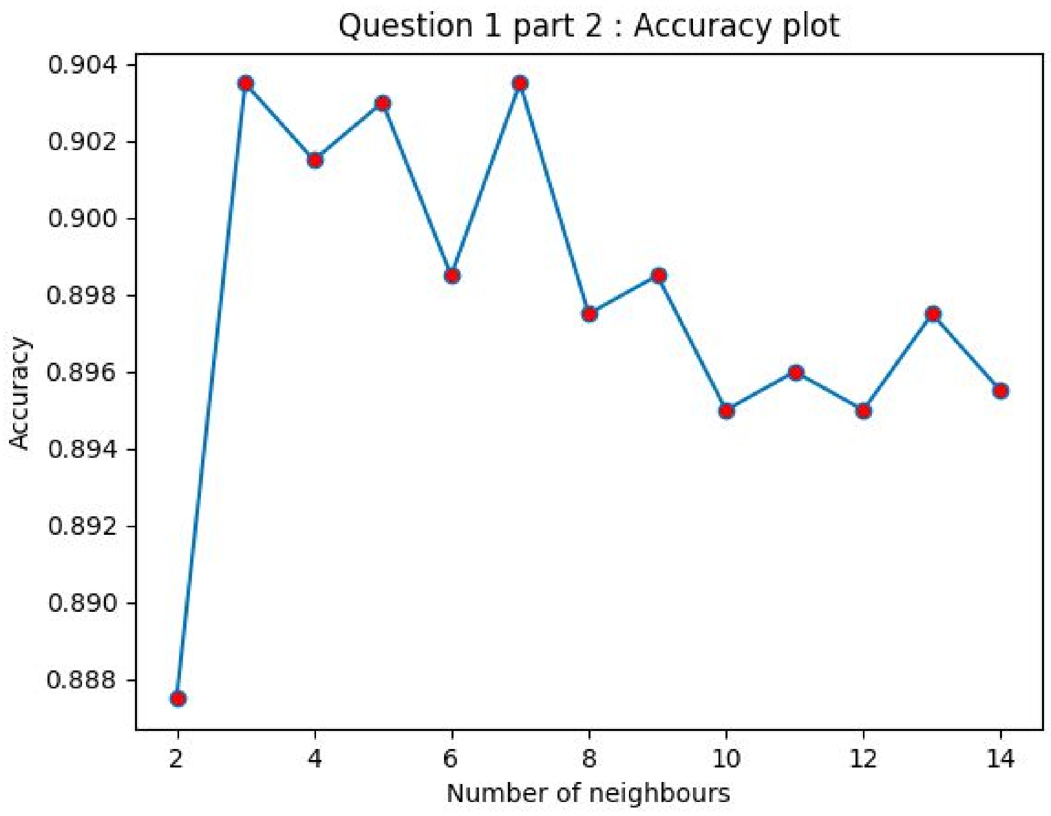
Question 1 : k Nearest Neighbour (kNN)

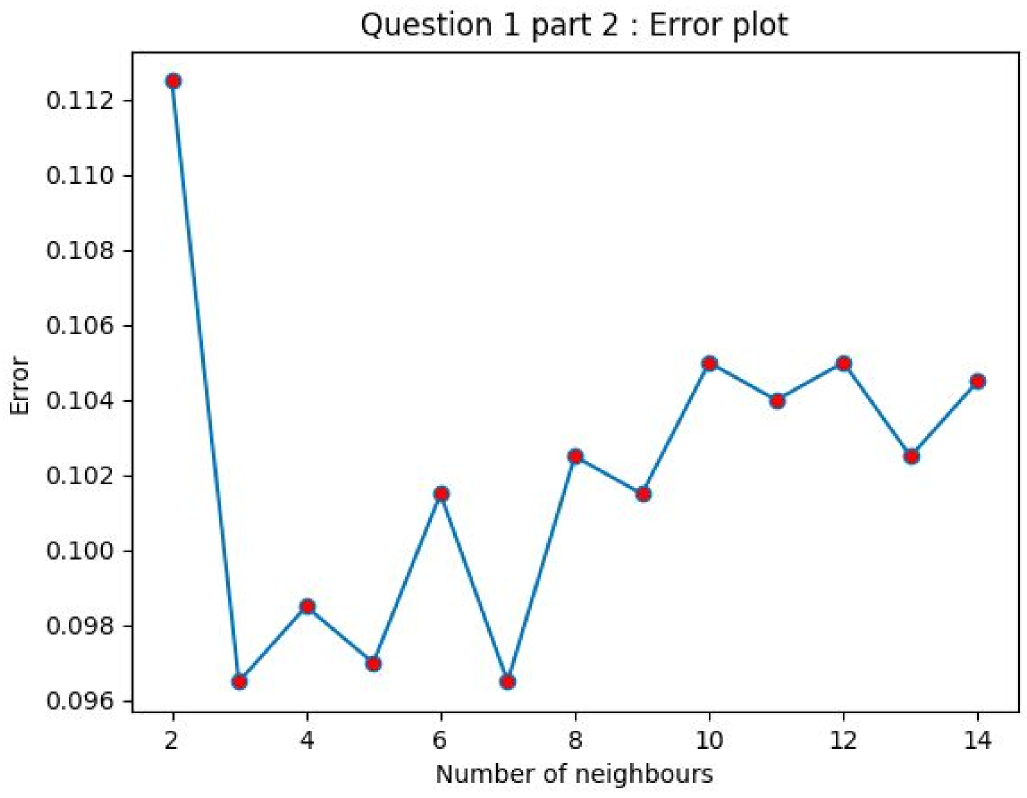
1) See file “Q1_1.py”. In this part we are supposed to load the data and visualize the data on a 2-d graph. The data the directly loaded from the file “sat.trn” (for training data) and “sat.tst” (for testing data). Data is reduced to 2 dimensions using TSNE. Below is the data visualization graph:



Next page ⇒

2) Below is the accuracy vs number of neighbors and error vs number of neighbors graphs :

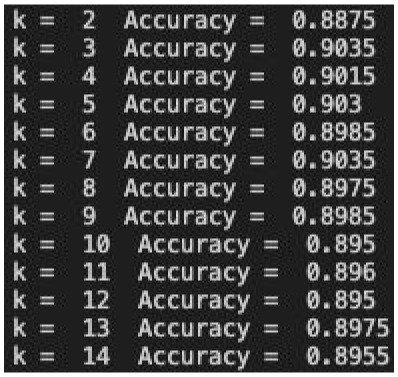




Since the accuracy plot makes a high at k = 3 and error plot makes a low at k = 3 it is very much clear that the optimal value for k (in kNN) algorithm is **3**.

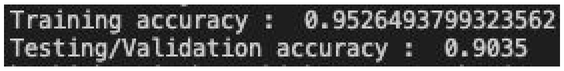
So, k = 3 is the optimal number of neighbours

3) Below is the validation accuracy from the function I have built :



```
k =  2  Accuracy =  0.8875
k =  3  Accuracy =  0.9035
k =  4  Accuracy =  0.9015
k =  5  Accuracy =  0.903
k =  6  Accuracy =  0.8985
k =  7  Accuracy =  0.9035
k =  8  Accuracy =  0.8975
k =  9  Accuracy =  0.8985
k =  10  Accuracy =  0.895
k =  11  Accuracy =  0.896
k =  12  Accuracy =  0.895
k =  13  Accuracy =  0.8975
k =  14  Accuracy =  0.8955
```

From the sklearn function (k = 3), following are the accuracy I am getting :



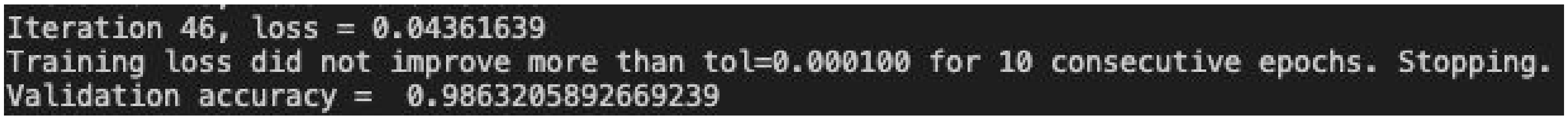
```
Training accuracy :  0.9526493799323562
Testing/Validation accuracy :  0.9035
```

The validation accuracy from the function I have built and from the sklearn function of kNN for k = 3 is **0.9035 = 90.35%.** My function is returning exactly the same value as the sklearn function.

Next page ⇒

Question 2 : Neural Networks

1) See file : ‘Q2_1.py’. Data is splitted into 80:20 ratios.
2) Following is the validation accuracy and loss after 46 iterations of the MLP :

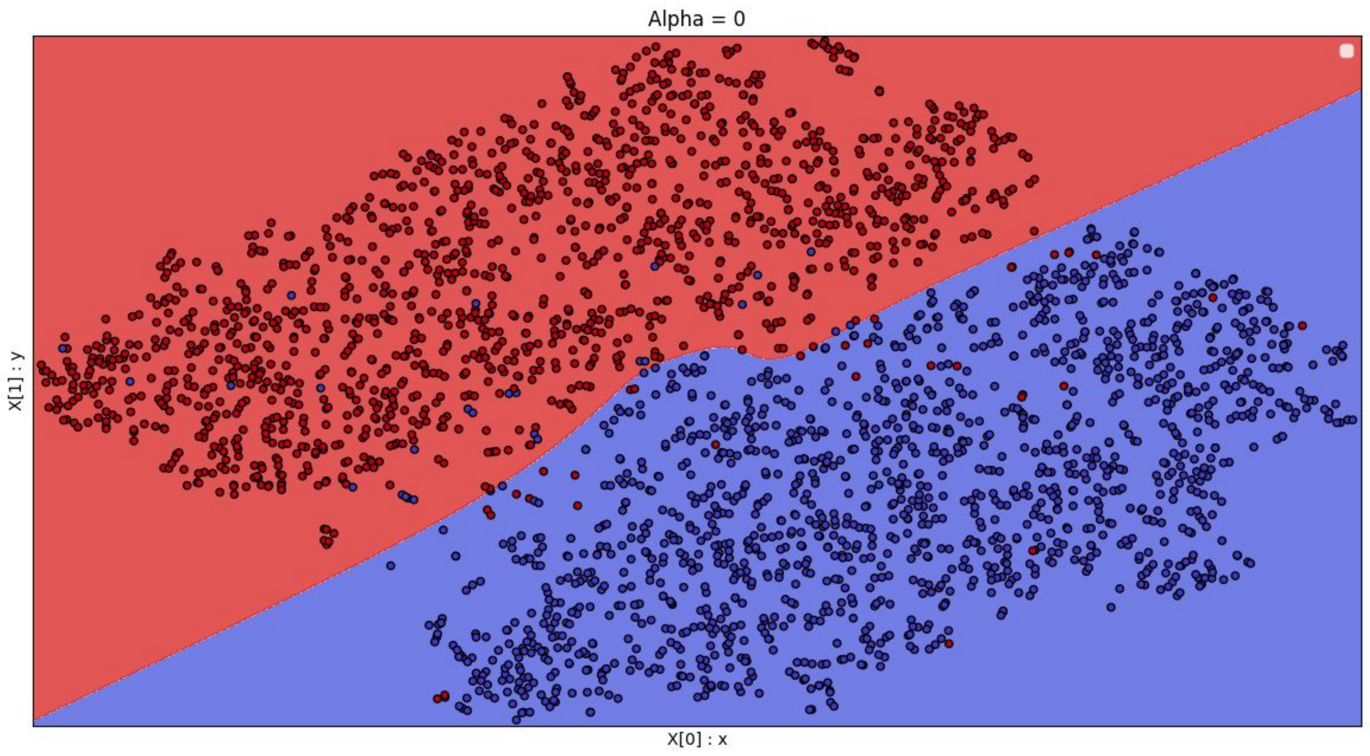


```
Iteration 46, loss = 0.04361639
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Validation accuracy =  0.9863205892669239
```

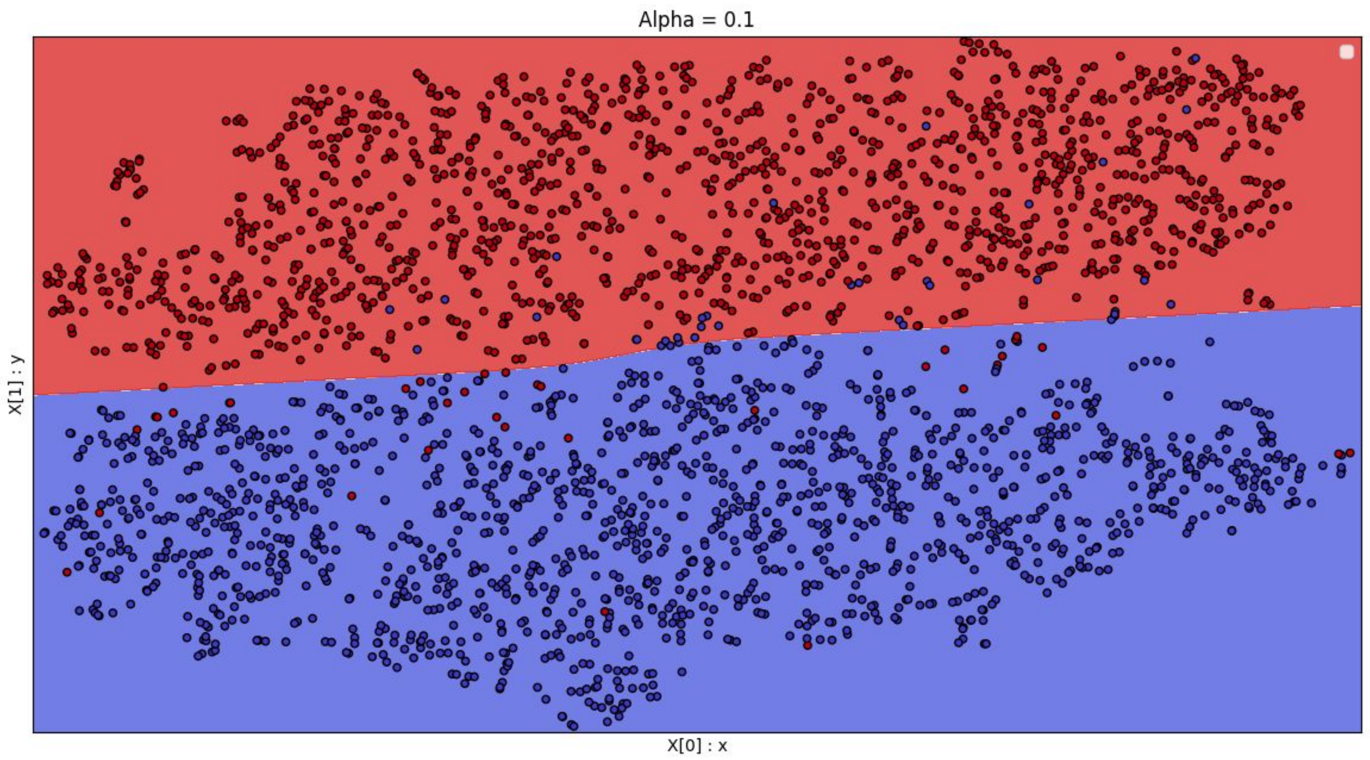
Loss = 0.04361 (after 46 iterations)
Accuracy on validation/testing data = 0.9863

3) Below it the decision boundary plot with different values of alpha. Alpha values used are **: [0, 0.1, 1]**

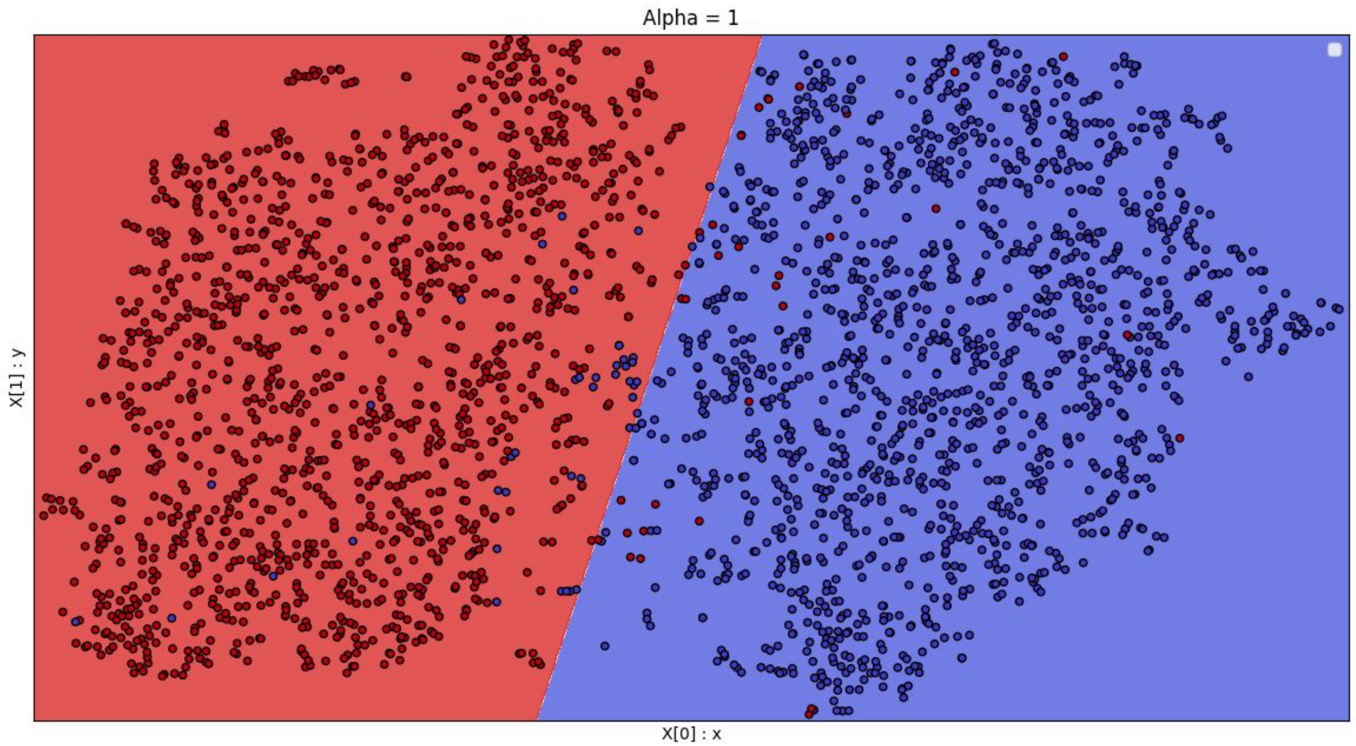
Alpha = 0



Alpha = 0.1



Alpha = 1


Alpha = 1
X[1] : y
X[0] : x