**Note: internet is required , data is downloaded from internet, Anaconda python 2 is required**

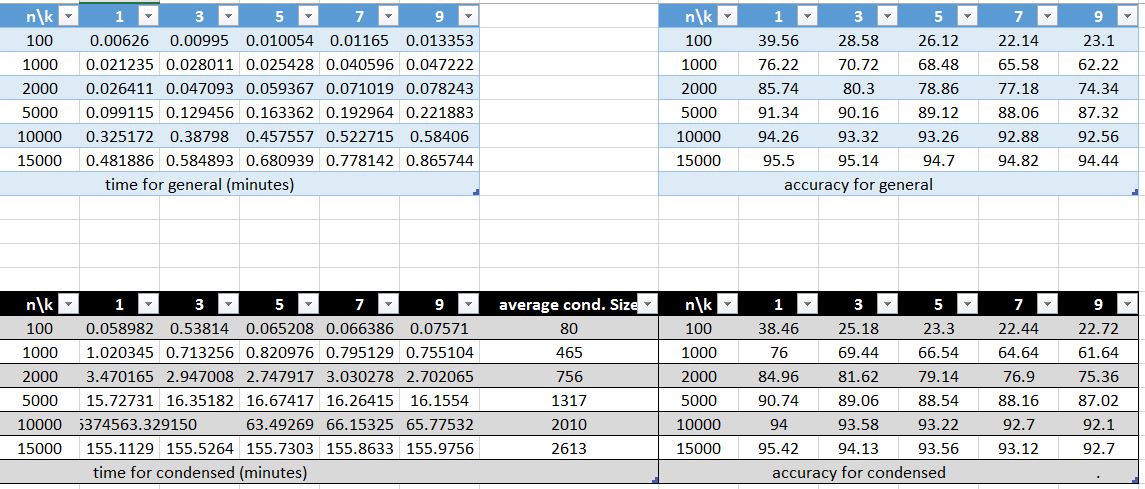
Algorithms Description

**Basic K-NN classification:** In this algorithm I am taking a sample from random dataset as training Data. For calculating distance between test point and all the training points I am converting the test point into Matrix of dimensions (training point,1) in each row I am using the same training point . Distance between all training points and test point is calculated using matrix operations. The nearest k neighbors y values are stored in the list. Then using counter we are taking the most repeated y value in the list. This value is the Testy value.

**Condensed Data:**  In this method we are reducing the training dataset into consistent set. Initially we take only one training point, remaining training points are took as test data and, randomly one point is taken for the incorrectly classified points and introduced into training data with correct class. This process is repeated till all the remaining training data is classified correctly. For each k value basic K-NN classification algorithm is called internally in the method for classifying the test data.

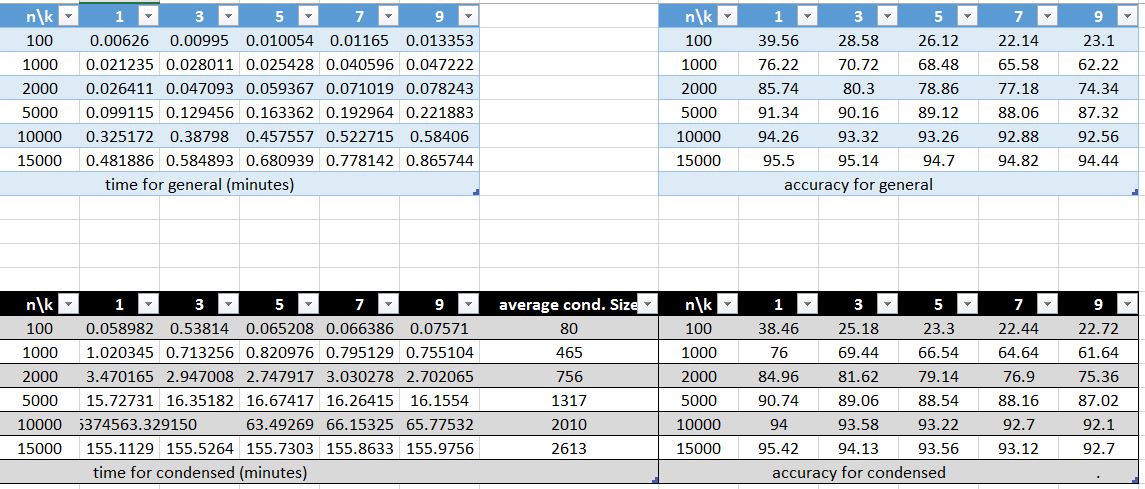
Running time and Classification Accuracy

**BASIC K-NN**

­­

Time and accuracy for basic K-NN is given, we can see that time is below 1 minute for all values

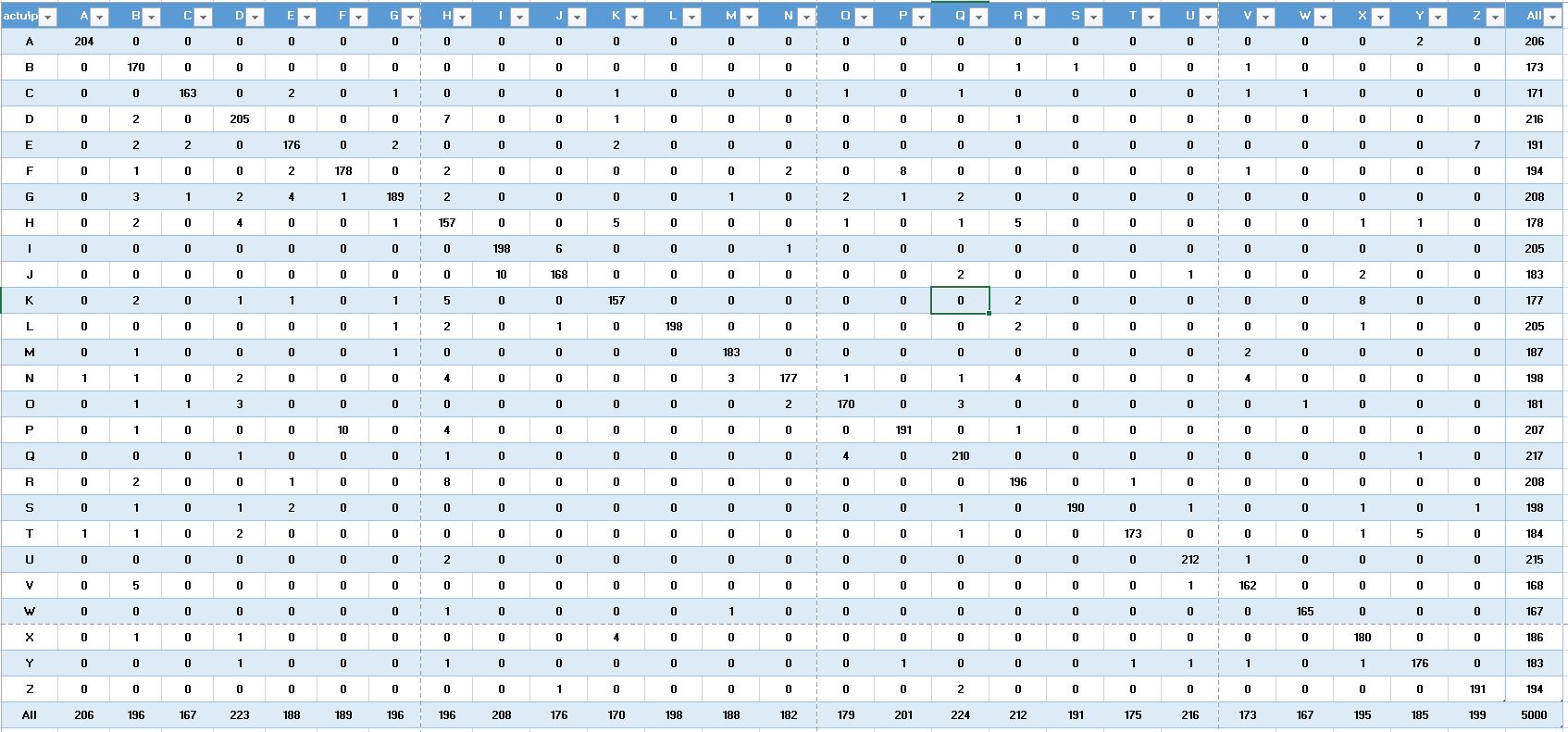
**CONDENSED K-NN**



The size of condensed training set is decreased to 15% of original training data.

**CONFUSION MATRIX FOR BASIC K-NN n=15000 K=5**

For brief lookup of confusion matrix open image in the folder

****

Changes made for Improving Accuracy and Time

* For high speed distance calculation between points, I have used numpy matrix, it has improved the time from 6 minutes to less than one minute
* For higher accuracy in condensed matrix, taking the random seed in all the iterations.