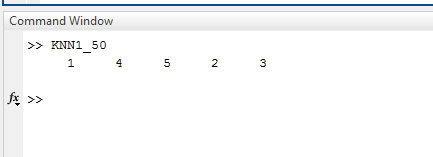
**KNN Classifier:**

Knn classifier is the lazy learning algorithm to output the class membership of given test data. Here, initially the train data and test data X are read and the output is the class prediction of the test data.

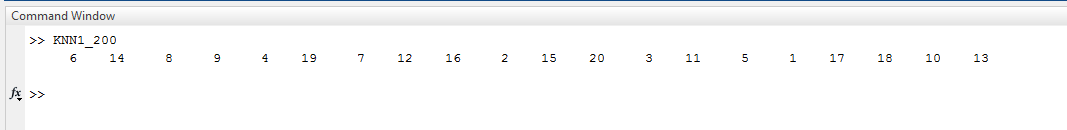
Euclidean distance is calculated using the formula D = abs(X1'.^2\*b - 2\*X1'\*Y1 + a'\*Y1.^2);

Based on the K value the majority votes of neighbour’s are considered to predict the class membership.

**Output for KNN 1\_ 50:**



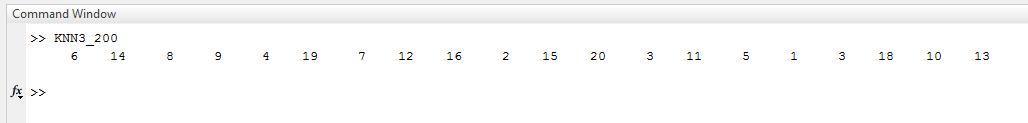
**Output for KNN1\_200:**



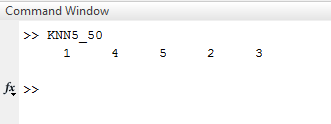
**Output for KNN3\_50:**



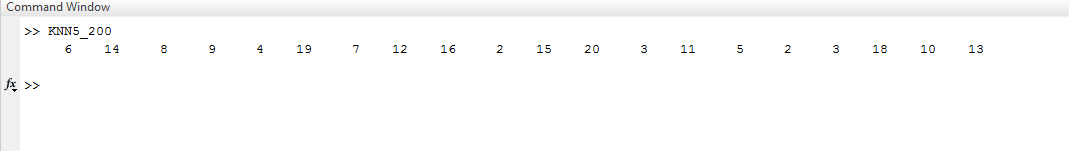
**Output for KNN3\_200:**



**Output for KNN5\_50:**



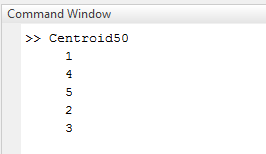
**Output for KNN\_200:**



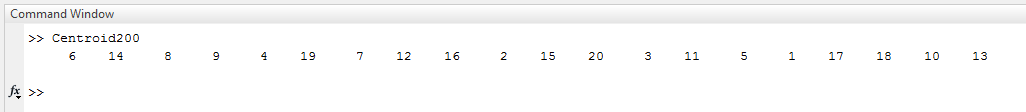
**Centroid Classifier:**

Centroid clustering algorithm also gives the prediction of the class membership of the given input test data. Means of the classes is calculated and the Euclidean distance is calculated to measure the nearest centroid of the cluster and the test data is assigned to that class. Also known as K-means clustering

**Output for Centroid\_50:**



**Output for Centroid\_200:**



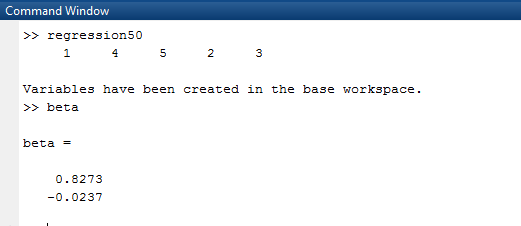
**Linear Regression Classifier:**

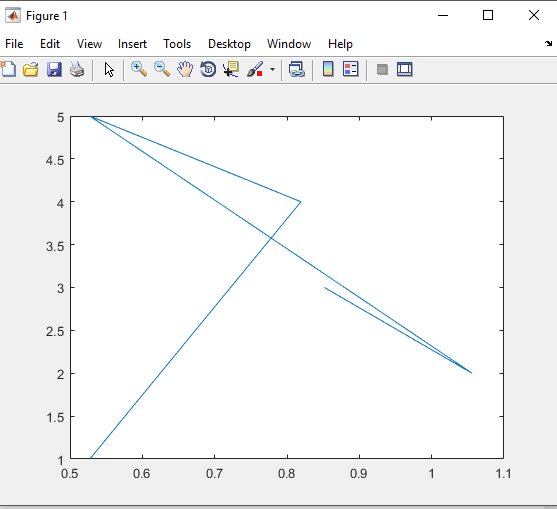
In this, Ytrain file is populated with the train data and fed to the given formula

B = pinv(Xtrain') \* double(Ytrain)' ; % (XX')^{-1} X \* Y'

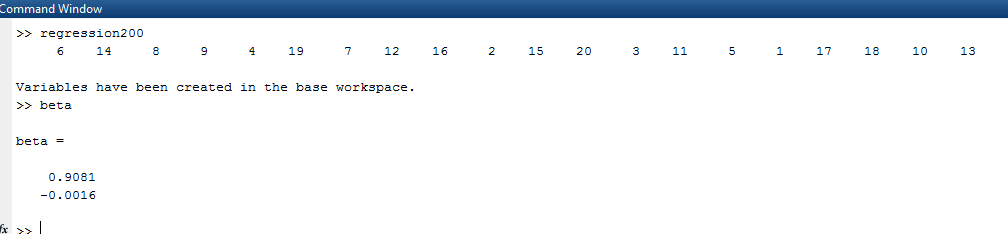
And the coefficients b0,b1 are calculated to represent the linear form of the line Y = b0 + b1X + C

**Output for Regression\_50:**





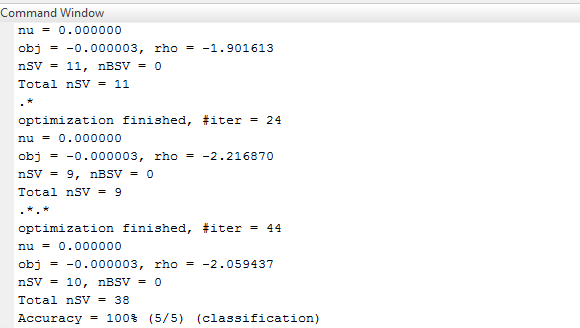
**Output for Regression200:**



**Libsvm Classifier:**

Initially, Input files are transformed into libsvm file format using libsvmwrite function. Svmtrain function is used to generate the model and the prediction of the classes for the test data is done by using the svmpredict function.

**Output for libsvm50:**



**Output for Libsvm200:**

