Question 2 – binary classification

For solving this question I have decided to use a linear SVM and a non-linear SVM.

# Linear method: linear SVM

For the implementation of linear SVM I have chosen to use to use the LIBSVM implementation for SVM and I have used the following functions from this library:

* *svmtrain* to build the model with the following parameters:
  + columns with parameters of train data and columns with classification for each pattern
  + kernel function chosen to be *‘linear’* in order to have a linear SVM
  + “*boxtconstraint*” set to be the C so I can specify the regularization parameter
  + “*autoscale*” set to be “*false*” prevents LIBSVM to do any normalization of data because I do that before using this function
* *svmclassify* which takes as input the model that was created previously and the test block to calculate predictions

For the model selection process I have gone through the following steps:

* I calculated mean and standard deviation for train data and used those values to normalize both: train and test dataset
* I applied cross validation to find the optimal value for C (the regularization parameter) and applied it for the following values: and as I did at previous exercises, I chosen to make 5-fold cross validation. The cross validation is applied inside the function *my\_fiveFoldCV* which takes as input the train data, the list of regularization parameters and the dimensions of blocks to make the split. The optimal value for C, which provides best accuracy, is returned.

The regularization parameter that offered the best accuracy during the cross-validation is C = 0.01 and used this value to build the model and calculated:

* accuracy for the train data: 0.9938
* accuracy for the test set: 0.9875
* sensitivity for the train set: 0.9938
* specificity for the train set: 0.9906
* sensitivity for the test set: 1
* specificity for the test set: 0.9705

As it can be noticed, values for accuracy, sensitivity and specificity are very good for both train and test set which means that the patterns with label 0 and 1 are well grouped and there is small interference between them in both train and test set. This could conclude that SVM with linear kernel is effective for this application: to detect the cybercrime.

# Non-linear method: SVM with non-linear kernel

For the implementation of non-linear SVM I have also chosen to use to use the LIBSVM implementation for SVM and I have used the following functions from this library:

* *svmtrain* to build the model with the following parameters:
  + columns with parameters of train data and columns with classification for each pattern
  + “*autoscale*” set to be “*false*” prevents LIBSVM to do any normalization of data because I do that before using this function
  + “*boxtconstraint*” set to be the C so I can specify the regularization parameter
  + kernel function chosen to be the kernel that I created: I have decided to use same Gaussian kernel as in the assignment 3:
* *svmclassify* which takes as input the model that was created previously and the test block to calculate predictions

For the model selection process I have gone through the following steps:

* I calculated mean and standard deviation for train data and used those values to normalize both: train and test dataset
* I applied cross validation to find the optimal value for C (the regularization parameter) and for kernel parameter. I applied the 5-fold cross validation for all the combinations between the following values: and. The cross validation is applied inside the function *my\_fiveFoldGaussianKernelCV* which takes as input the train data, the list of regularization parameters, kernel parameters and the dimensions of blocks to make the split. The values of C and which show the best accuracy are returned by the function.

The regularization parameter that offered the best accuracy during the cross-validation is C = 10 and the best Gaussian parameter is 0.001 .Using these values I built the model and calculated:

* accuracy for the train data: 0.9953
* accuracy for the test set: 0.9875
* sensitivity for the train set: 0.9969
* specificity for the train set: 0.9938
* sensitivity for the test set: 1
* specificity for the test set: 0.9750

As it can be noticed, values for accuracy, sensitivity and specificity are also good for using the SVM model with Gaussian kernel, actually slightly better than using the linear one.