This project was completed in order to demonstrate how spoofs work for Bayes classifer. In this project, The data was imported for Gelatine Train and Test, Live Train and Test, and Latex Train and Test matrices. The Gelatine and Latex were fake results for fingerprints, but the Live data was for real fingerprints.

For the first part, the Naïve Bayes classifer was trained using the Latex and Live training set (LatexTrain and LiveTrain in Matlab code). The target variables were also generated which were known as labelsLiveLatex (1200x1 matrix), using the class names: ‘fake’ and ‘real’. The target variables ‘fake’ were assigned to Latex data and ‘real’ was assigned to Live data. The training features data was concatenated using the vercat() function in Matlab using the matrix for LatexTrain and LiveTrain. These variables were then used to generate the model, which is known as Part2Model using the fitcnb() function. For part 2, the Latex and Live model (Part2Model) was tested with the LatexTest and LiveTest data. This was done using the predict() function in matlab. The matrix results for this are known as LatexTestPredictPart2 and LiveTestPredictPart2. For Part 3, the Latex and Live Test Labels were created for the loss evalution. These were known as LiveTestLablesAdj and LatexTestLabelsAdj. The resuts for the losses were 0.097 for Live (lossLivePart3) and 0.5350 for Latex (lossLatexPart3). The Live loss was reasonable, but the Latex loss was absurdly high. What I believe might have caused it was that the model had too much Live training data and not much Latex training data, which was conducted having a ratio of 5:1 of Live to Latex. As a result, I think that the classifier model might have predicted some of the Latex test data to be ‘real’ when it was actually fake. Therefore, I belive there needs to be more varied data in the training to give a more ideal classifer. The resubstitution error was 0.1000 (resubLatexLivePart3), which was pretty reasonable. This was done using the resubLoss() funciton in Matlab.

For the fourth part, the goal was to create another classifer model. This time using Live, Latex, and Gelatine training data. The procedure of this was similar to how the Latex and Live model was created in Parts 1-3. This model was designated in the code as Part4Model. The predicted results for Live and Gelatine are shown as LivePredictPart4 and GelatinePredictPart4 in the code. As for the loss results, the Gelatine loss (lossGelatinePart5) was 0.0900 and the loss for Live (lossLivePart5) was 0.2790. The Gelatine loss was pretty reasonable. The Live loss was kind of high, but not as bad as the Latex loss in Part 3. Speaking of Latex, I actually calculated the Latex loss with the Part4Model and the loss was 0.1550 (lossLatexPart5), which is significantly better than the Part2Model, which was 0.5350. As a result, this shows that the Part4Model was a better classifier than the Part2Model, which only had Latex and Live training data. What I believe allowed for this improvement, was that the Part4Model had more varied data with the Gelatine being added. With more variances in training data of what can be fake, it can allow for more accurate results when determining what is fake than from what is real. This makes sense when considering the Occam’s razor theorem of more varied data. The resubstittuion error was 0.1736 (resubGelatineLivePart5), which is very legitimate.

For the sixth part, the goal was to train the classifier from the Part 1-3 but specifiy the priors to be 0.6 for Live and 0.4 for Latex. This model was called Part6Model. Apart from the prior, the Matlab code followed the same procedure as the previous parts. What was interesting was that the loss increased for Live, from 0.0970 (lossLivePart3) to 0.1450 (lossLivePriorPart6), and it decreased for Latex from 0.5350 (lossLatexPart3) to 0.5050 (lossLatexPriorPart6). I feel the results would have been opposite because the prior increased for Live, which means that it is more likely that it will be Live when it is assumed to be Live based on Bayeis Decision Rule. But it looks like the classifier is guessing more Live tests are fake compared to the Part2Model. The same argument can be said about the Latex results. As a result, I thought that was somewhat interesting. The resubstitution error was 0.1078 (resubLossPriorPart6), which again seems reasonable.

**Results from Matlab Workspace**

