Sahil Gandhi

Programming Assignment - 02

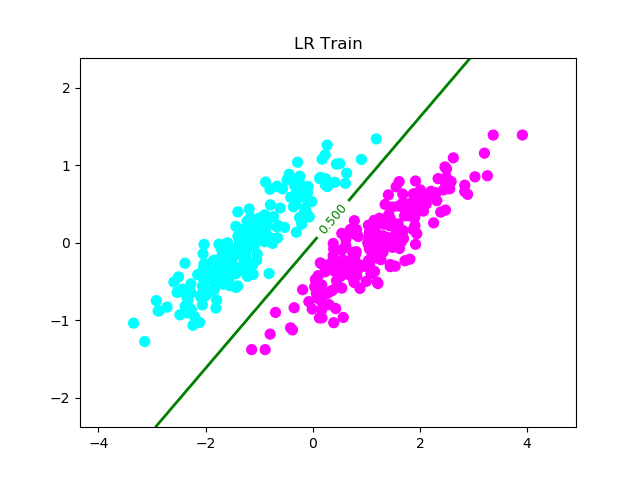
**Q1. Logistic Regression**

Part b. Test your implementation on the data sets provided

**Linearly Separable data**

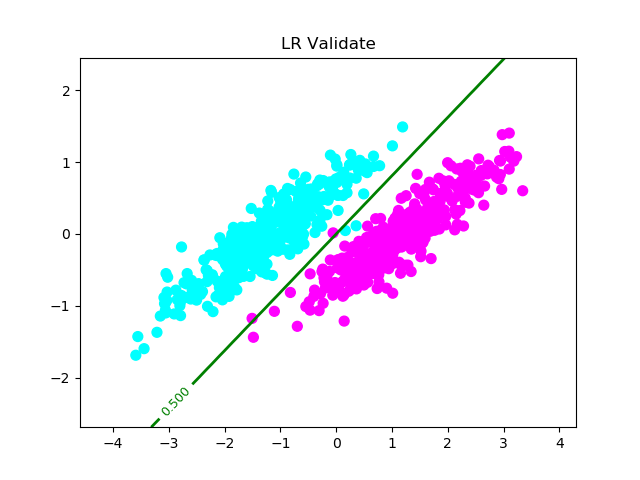
Training data:

* Misclassification on training data = 0



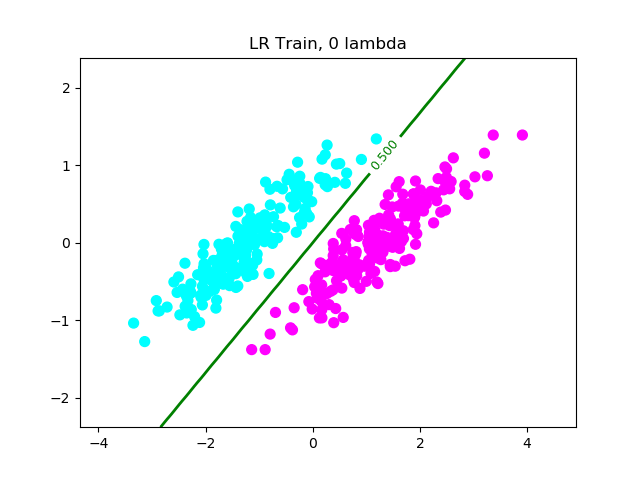
Validation data:

* Misclassification on validation data = 4



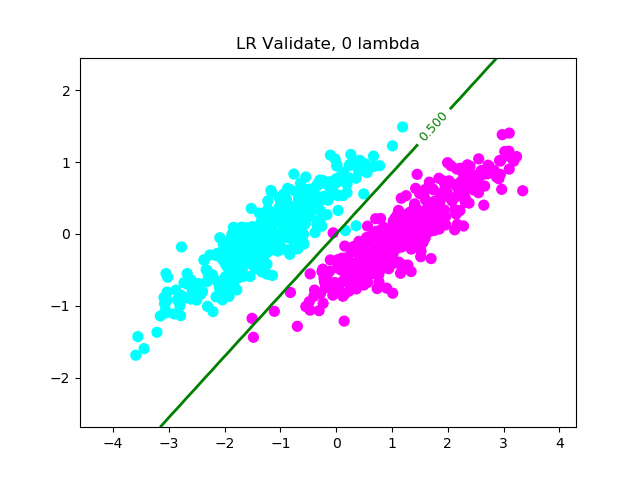
Training data:

* Misclassification on training data = 0



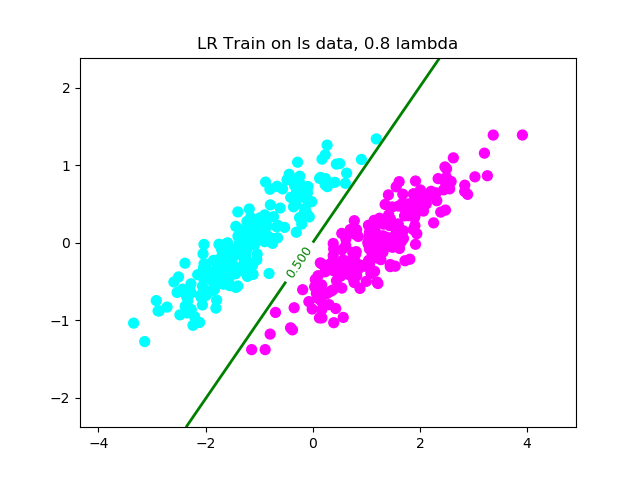
Validation data:

* Misclassification on validation data = 4



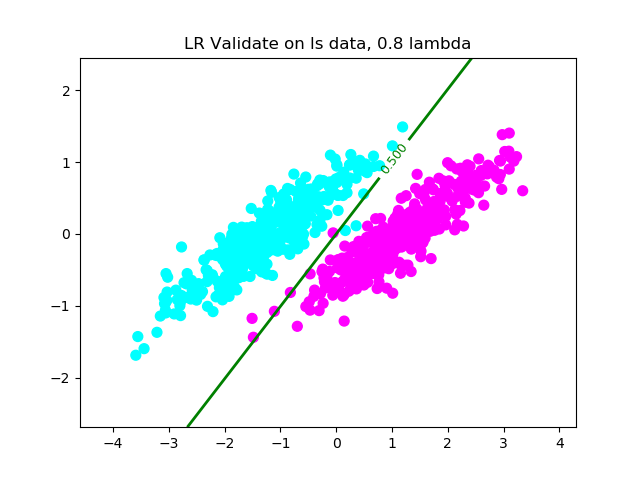
Training data:

* Misclassification on training data = 0



Validation data:

* Misclassification on validation data = 7



Training data:

* Misclassification on training data = 7

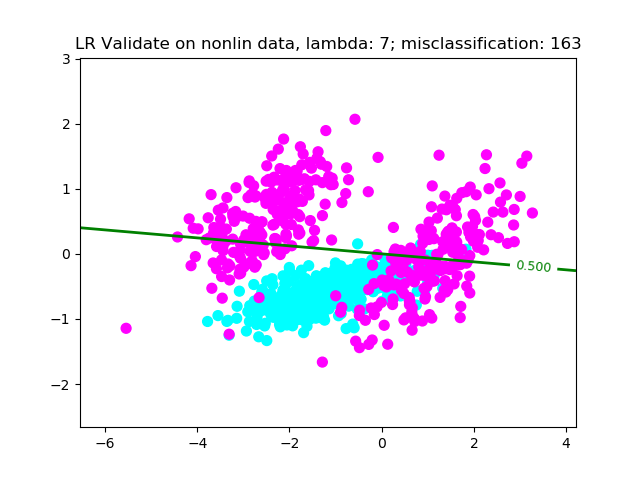
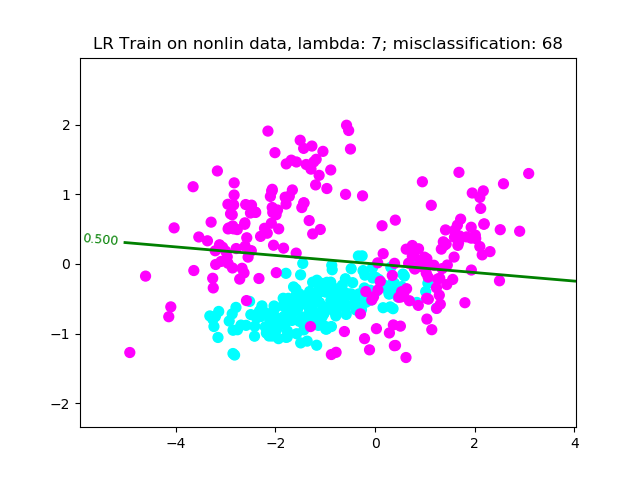
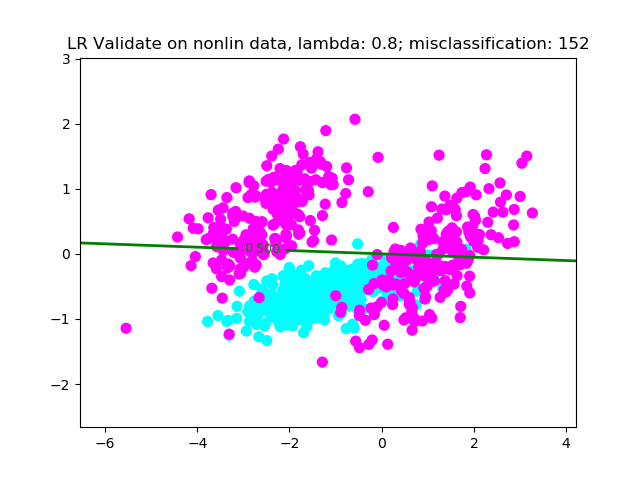
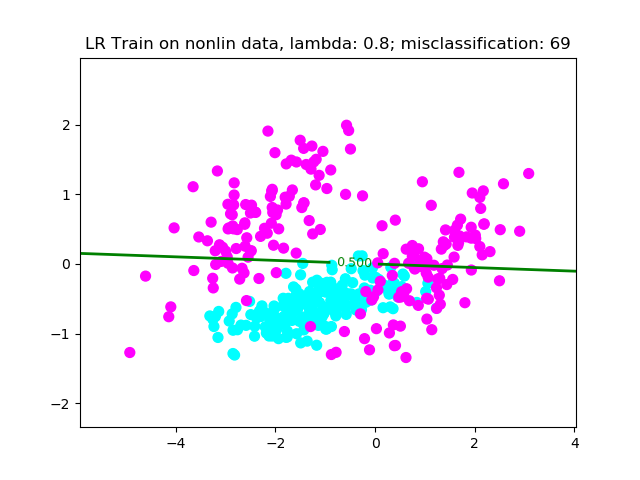
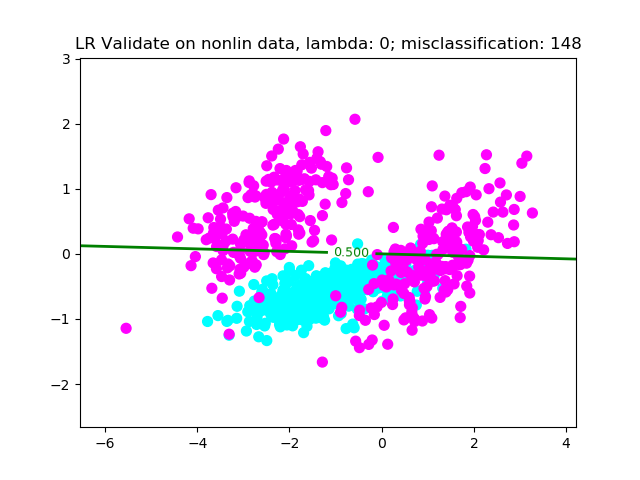
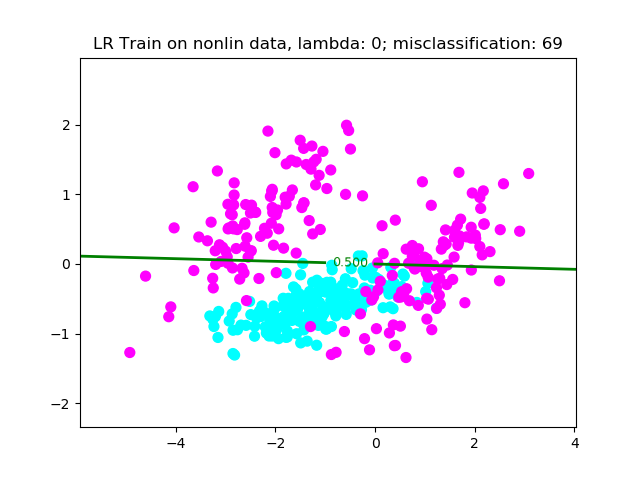
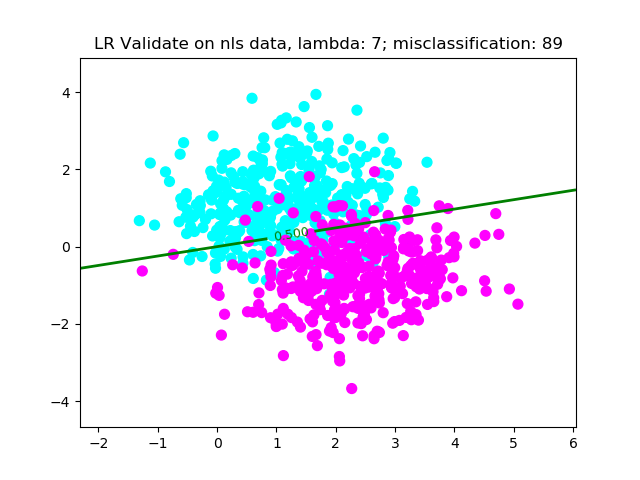
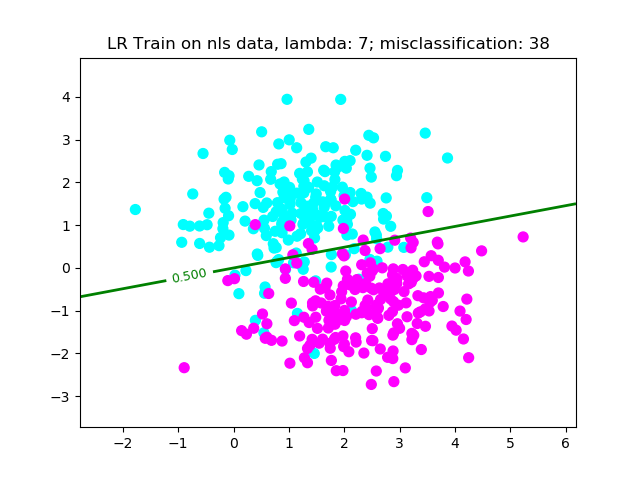
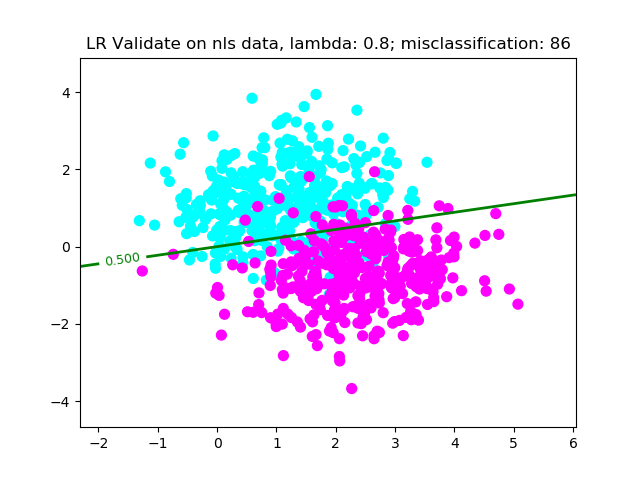
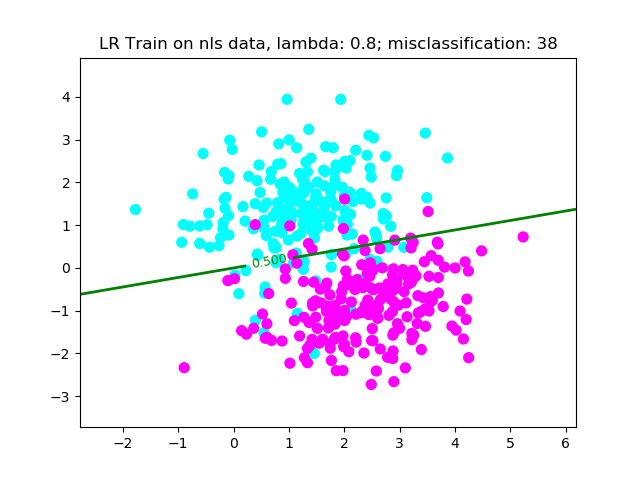
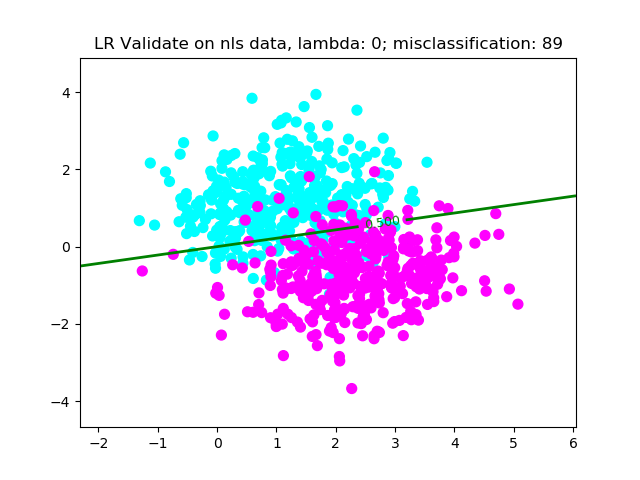
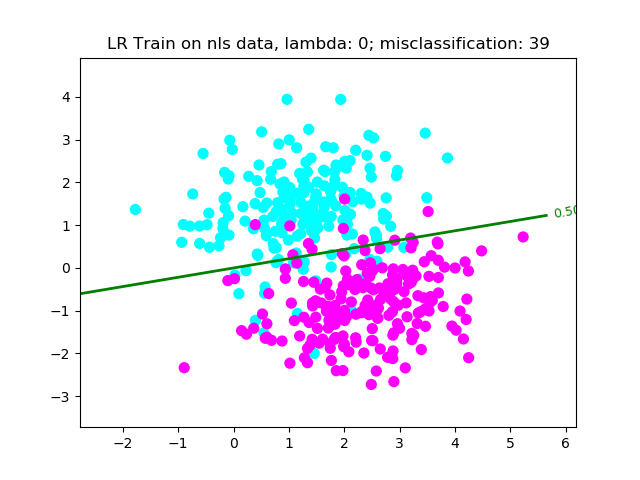


Validation data:

* Misclassification on validation data = 15



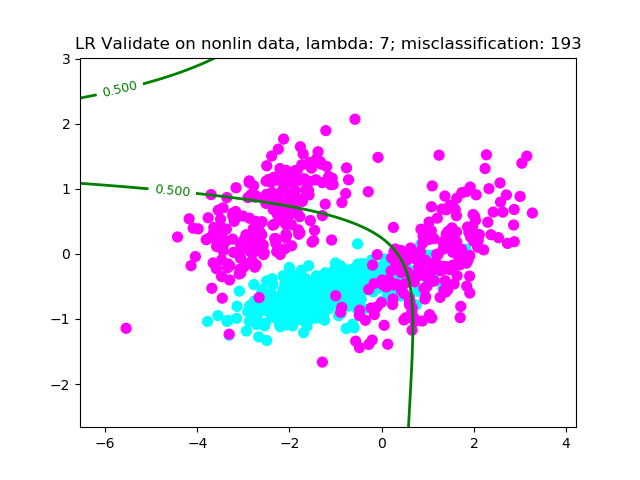
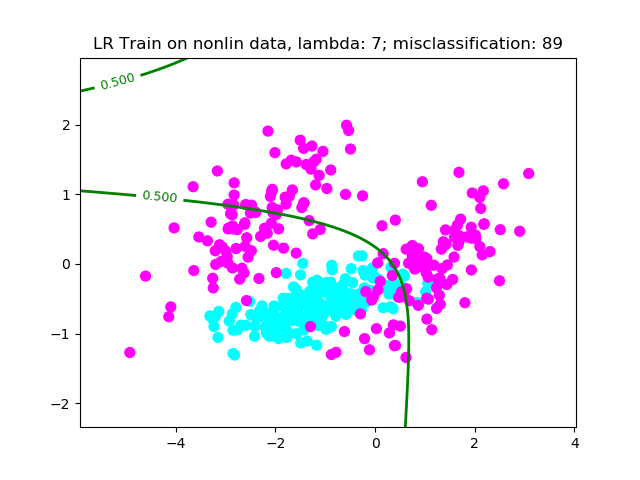
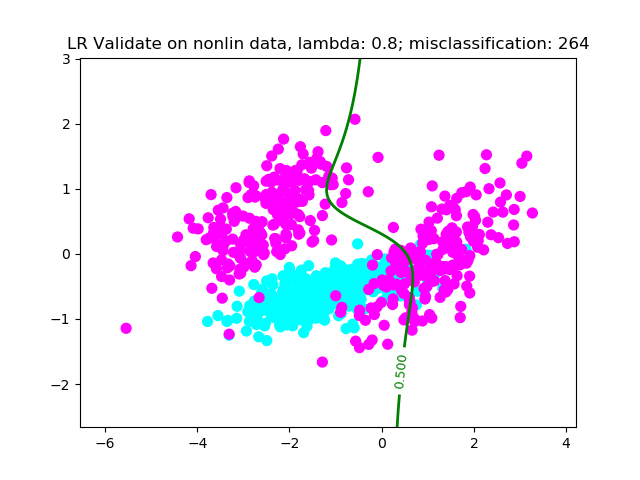
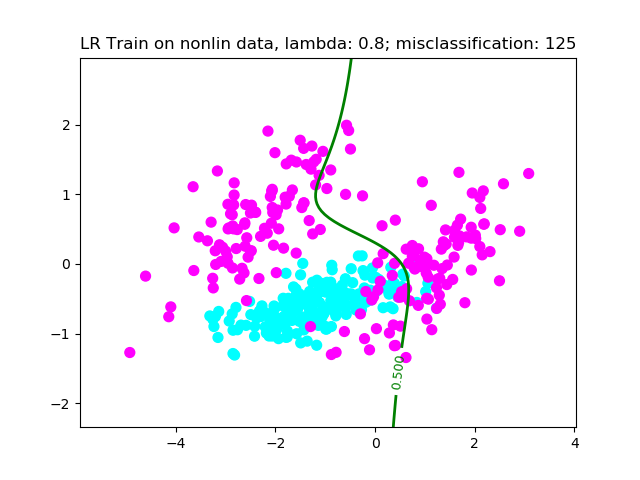
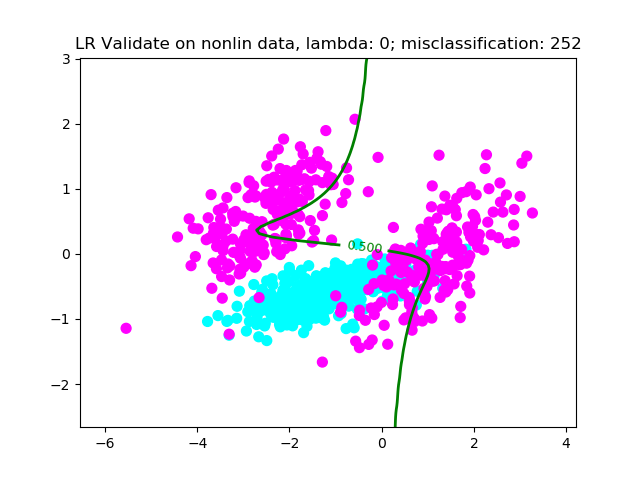
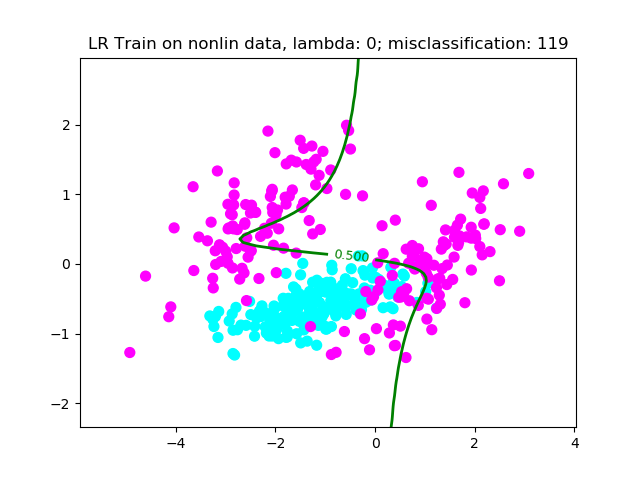
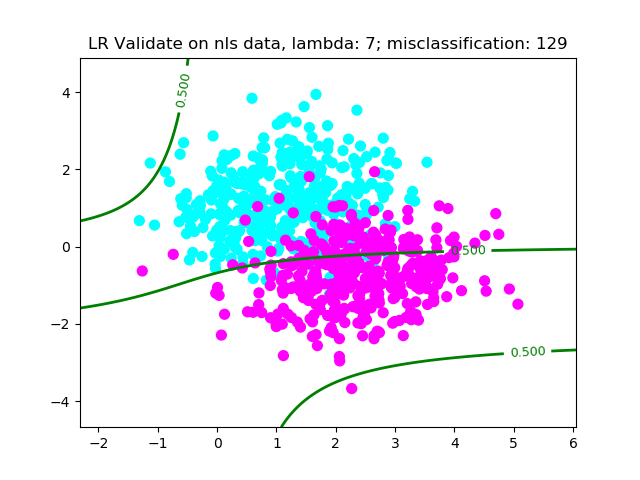
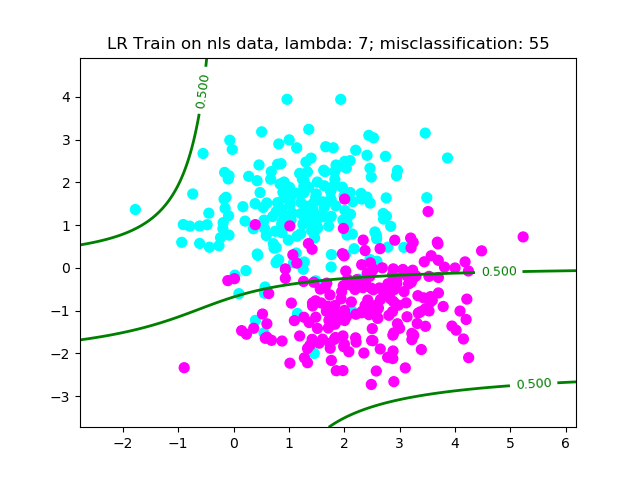
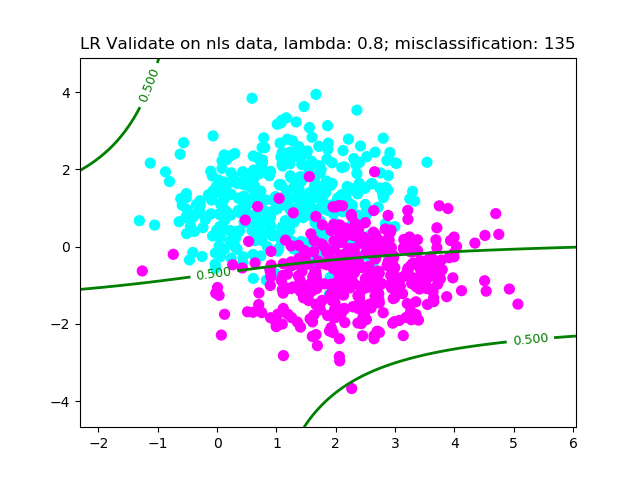
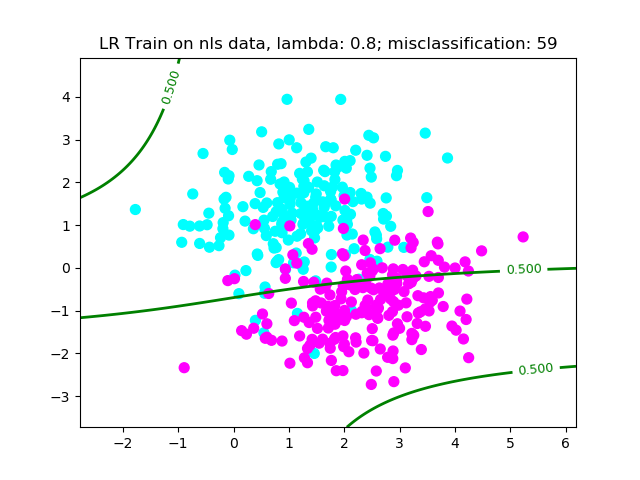
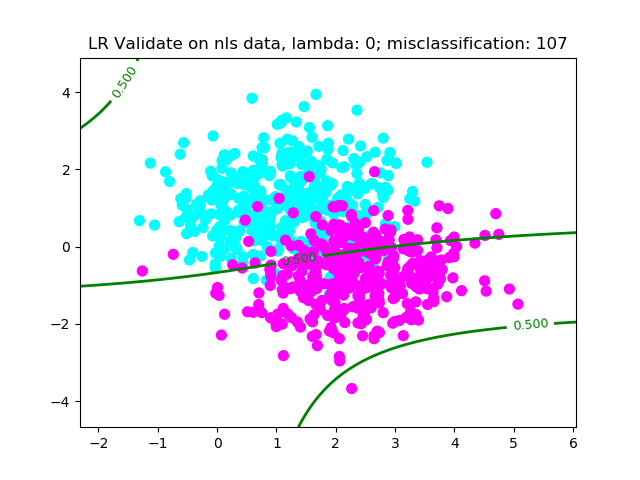
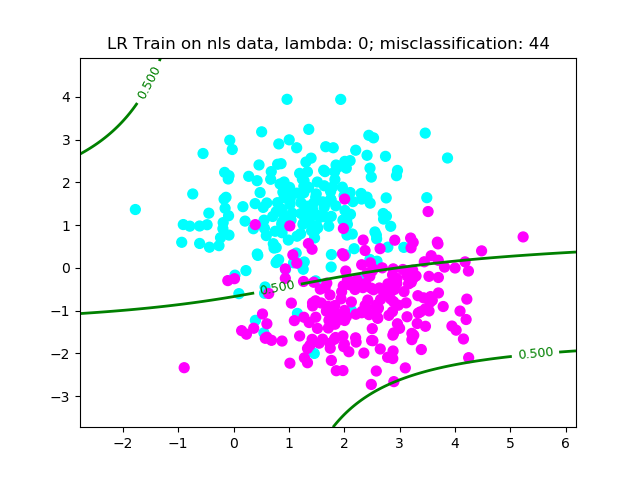
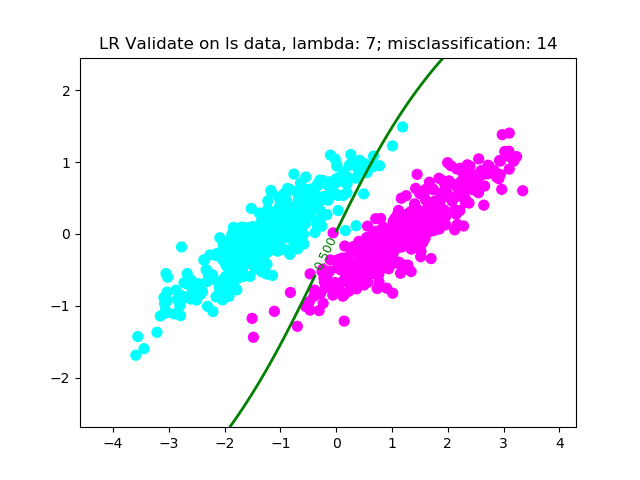
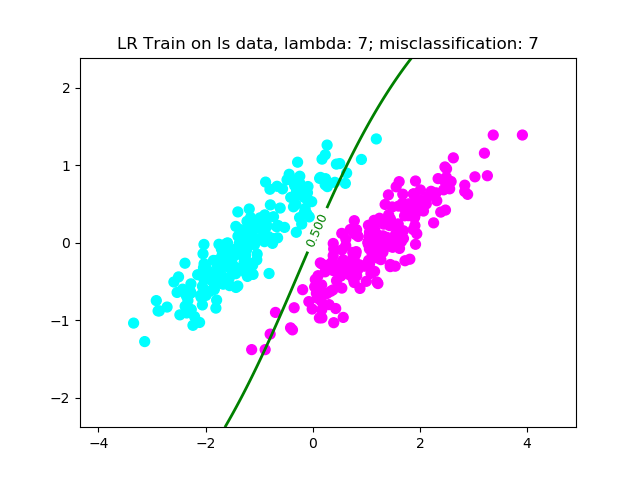
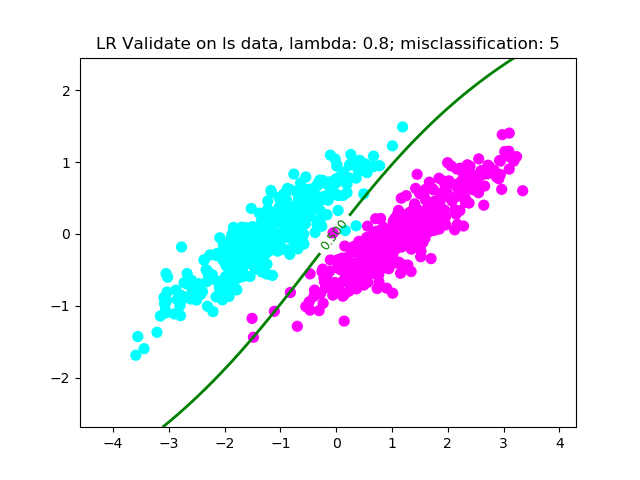
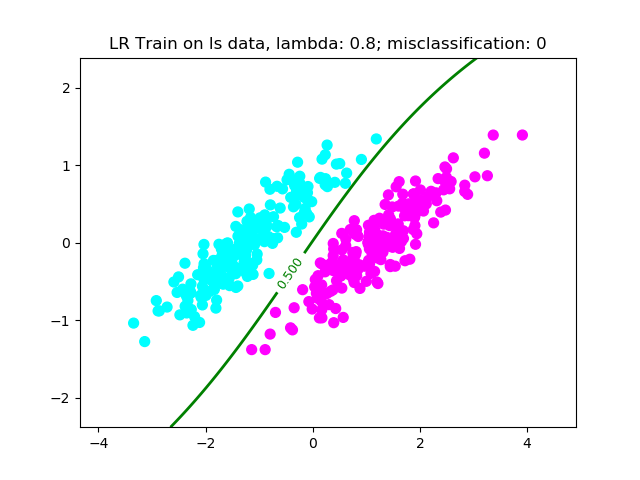
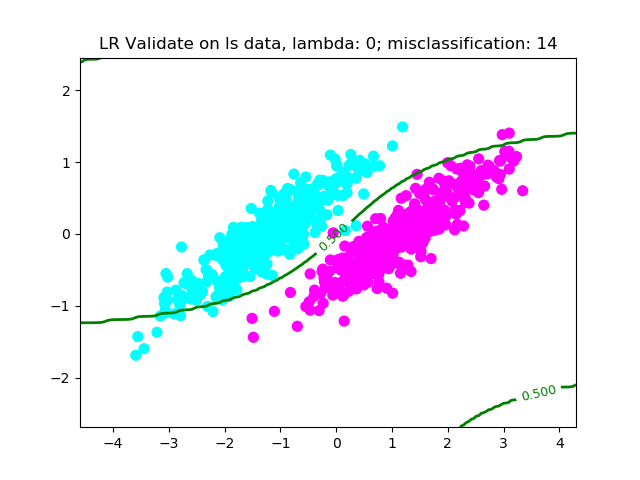
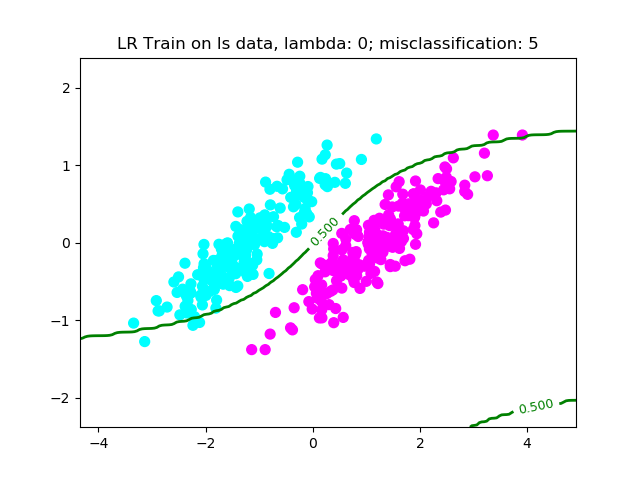
Below are plots by running logistic regression on other data sets with varying lambda. The misclassification count and lambda are in the plot title.



Analysis of all the plots in Q1. Part b.

* Increasing penalizes the shift in the weight vector and hence, in linearly separable data, it gets away with only a few misclassifications, but in other datasets, the misclassifications are very high.
* For non-linearly separable data, the model tries to reduce the overall error. The classification is not perfect, (obviously) but the chosen decision boundary has least possible misclassifications.

**Part 3. Second order polynomial basis functions**

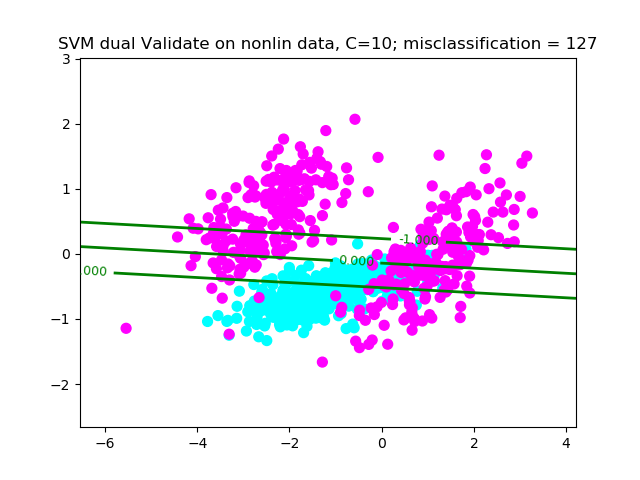
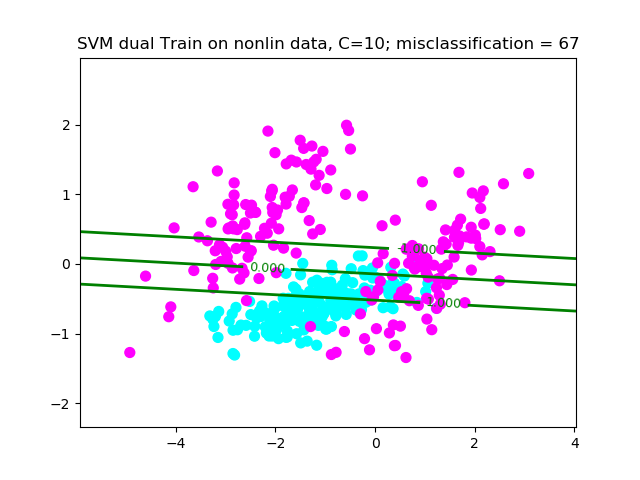
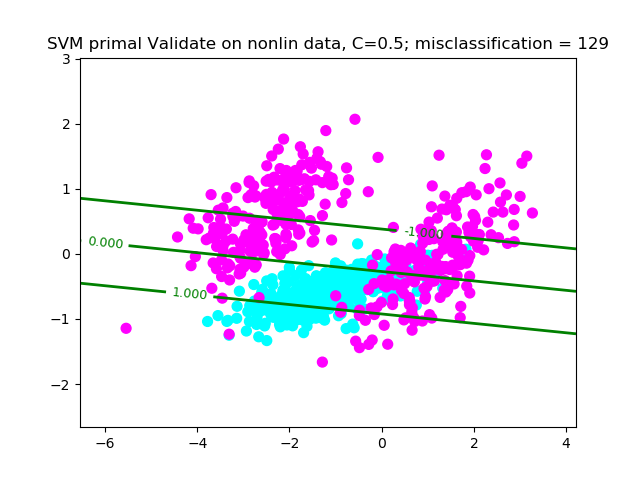
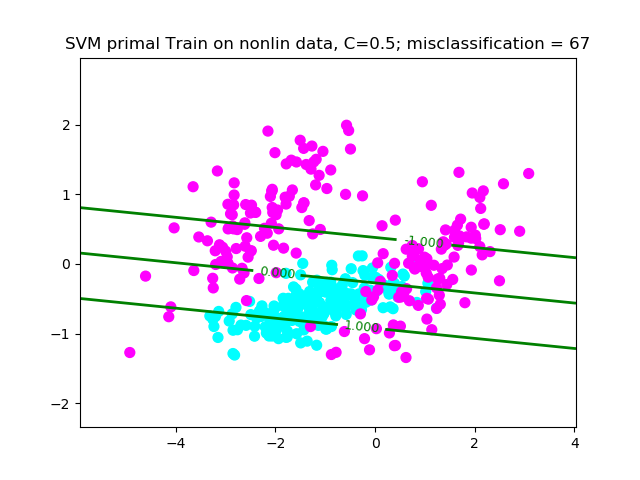
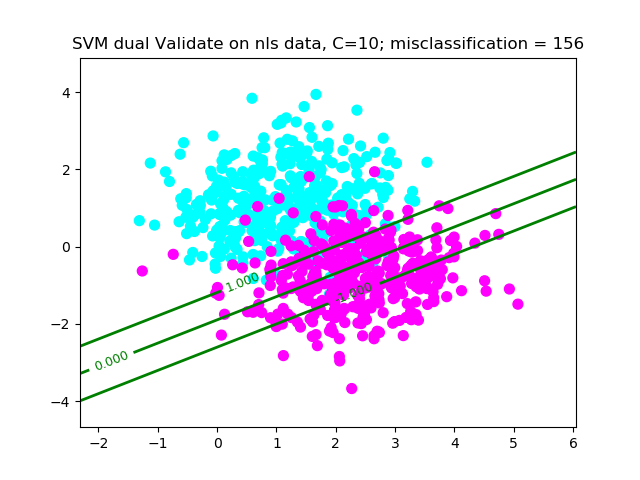
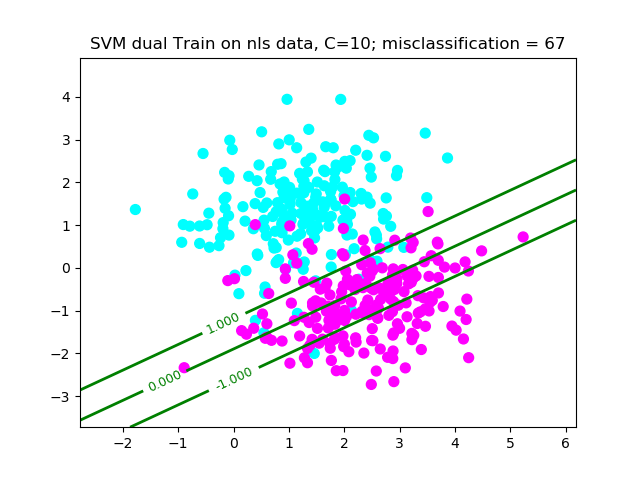
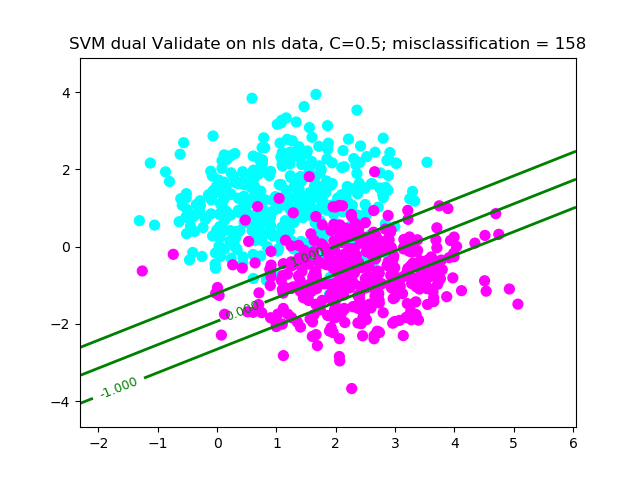
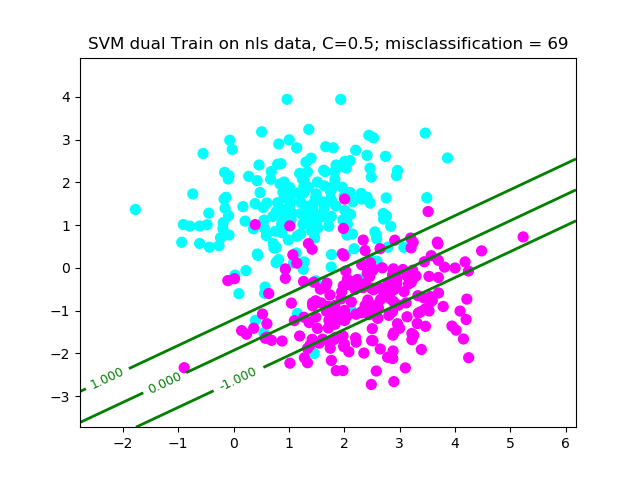
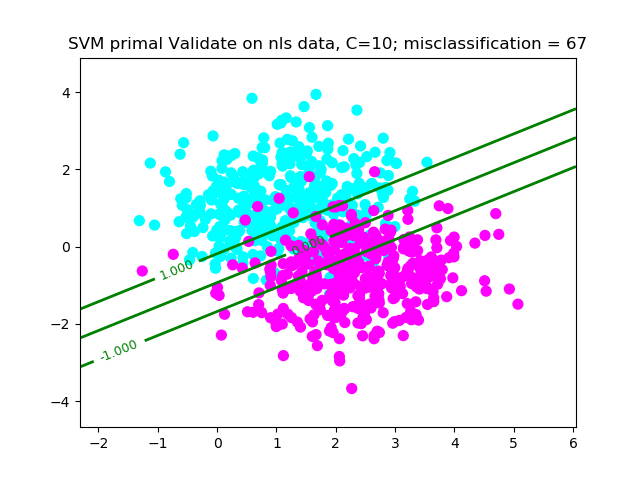
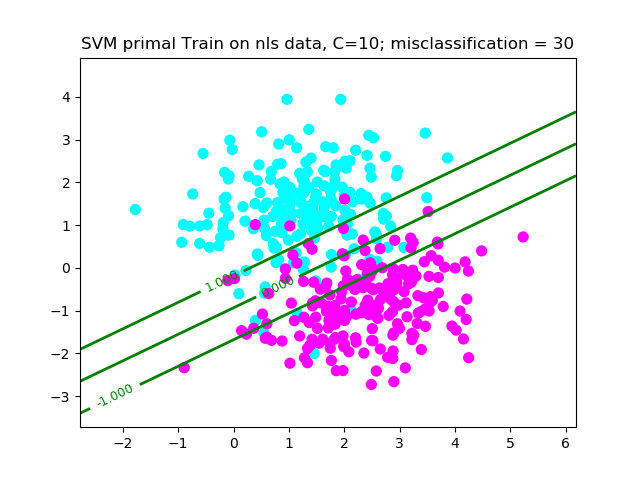
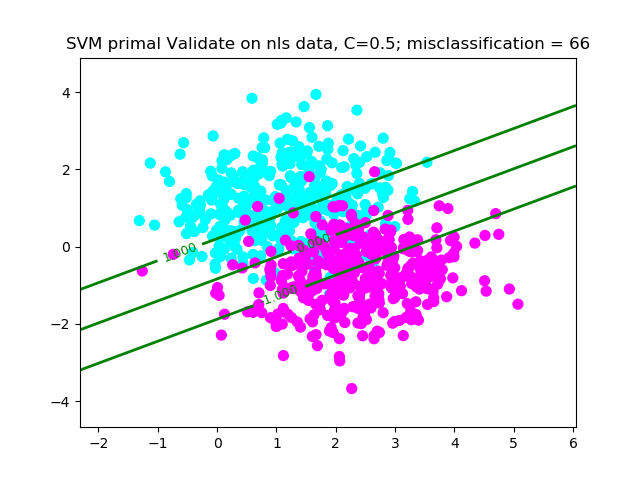
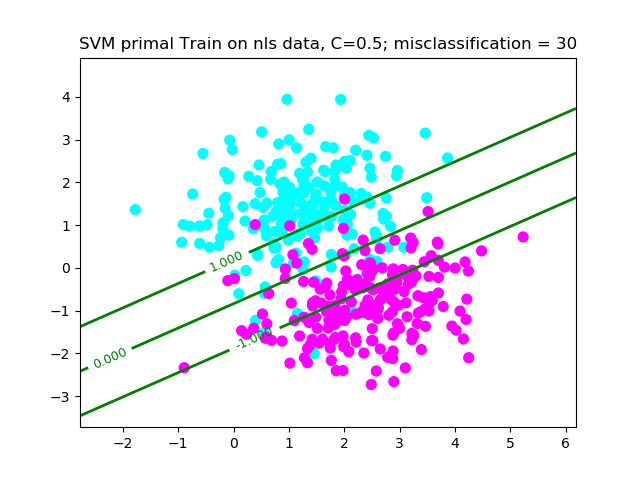
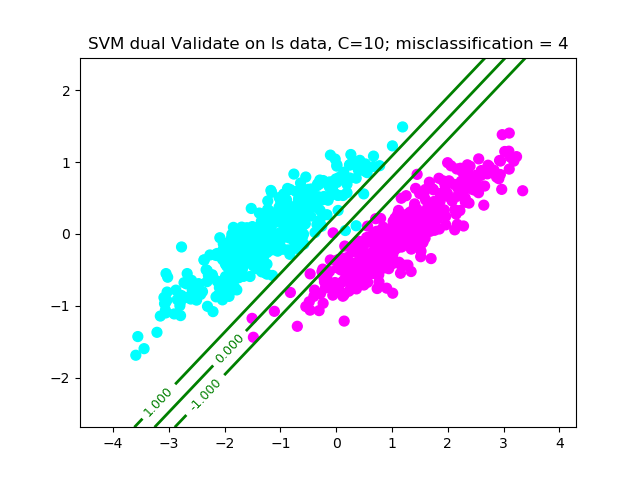
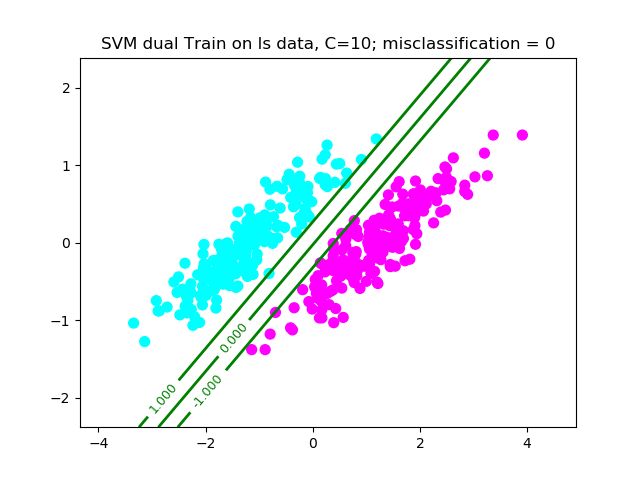
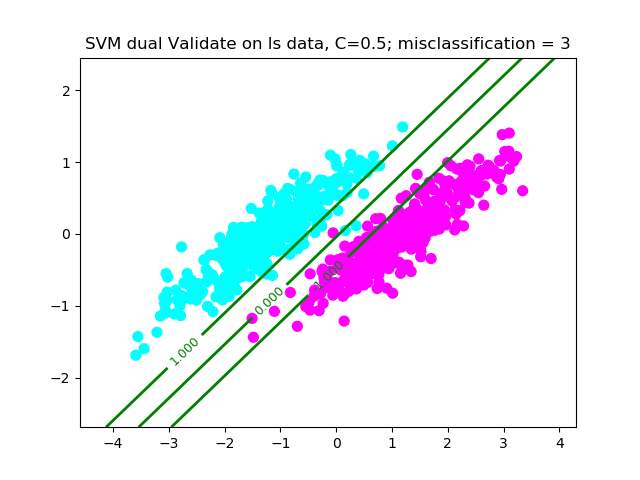
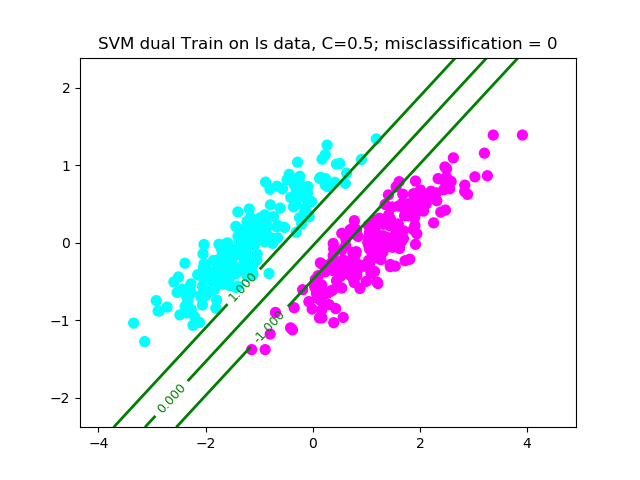
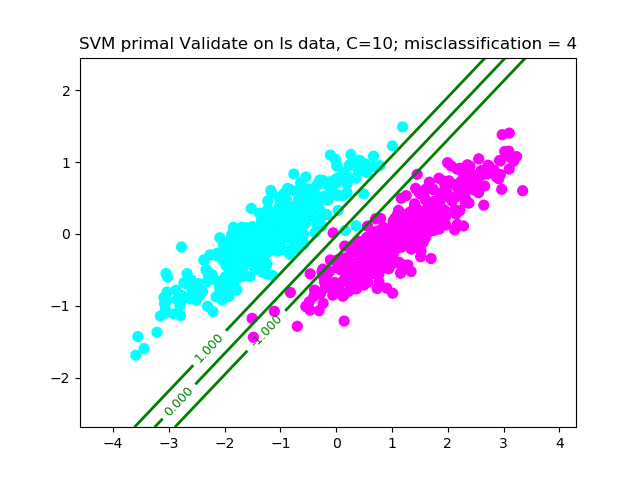
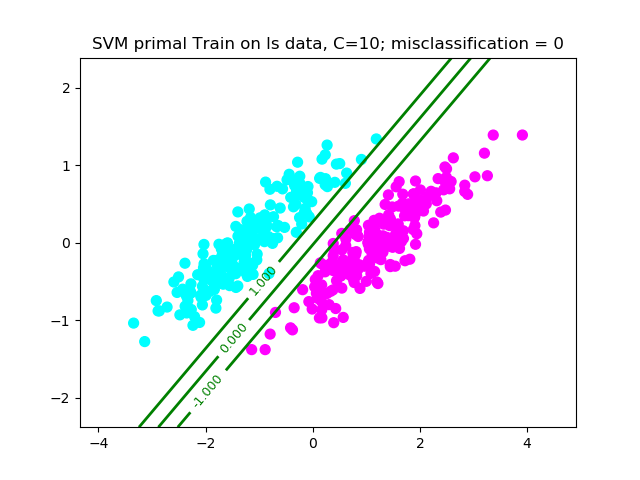
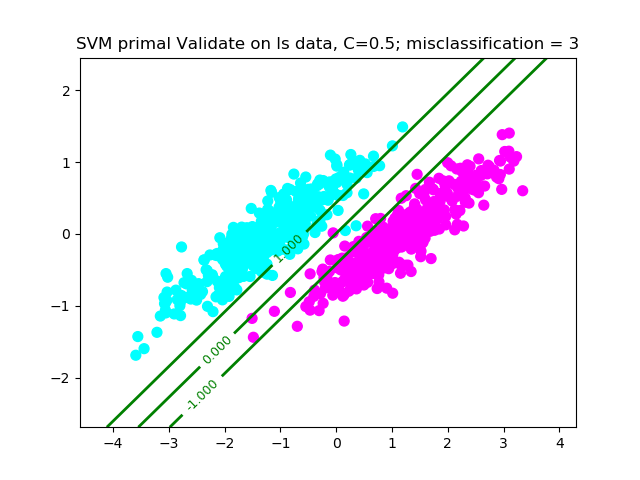
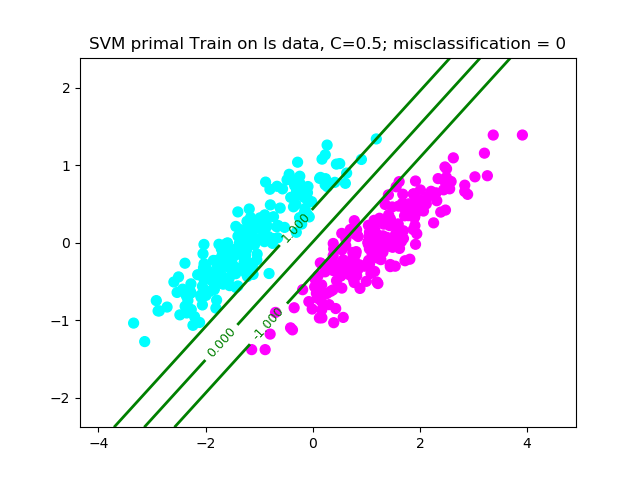


Analysis:

* As we can see, generating features in higher dimensions is a better model for non-linearly separable data at least for the training sets.
* Fine tuning lambda is required, as in all out above reports, the miscalculations on the training set was 0, but had very high miscalculations on validation sets; this clearly means, that the model is overfitting the training data.
* We can choose the lambda by running various values on the validation set for all models generated from the training set.

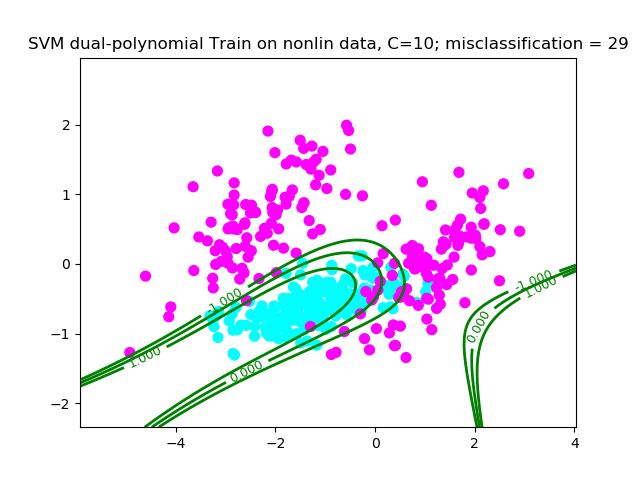
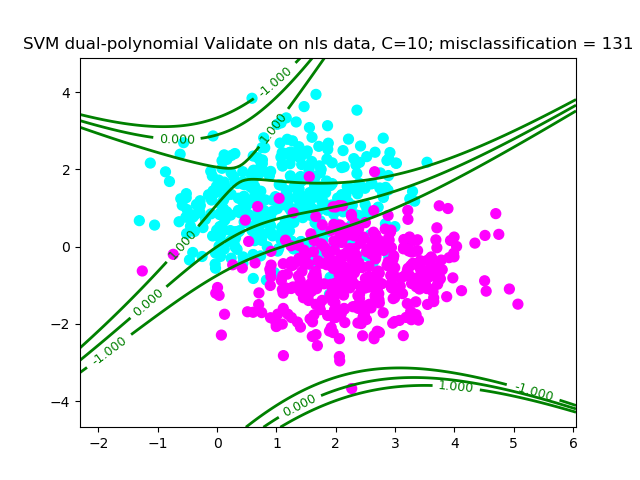
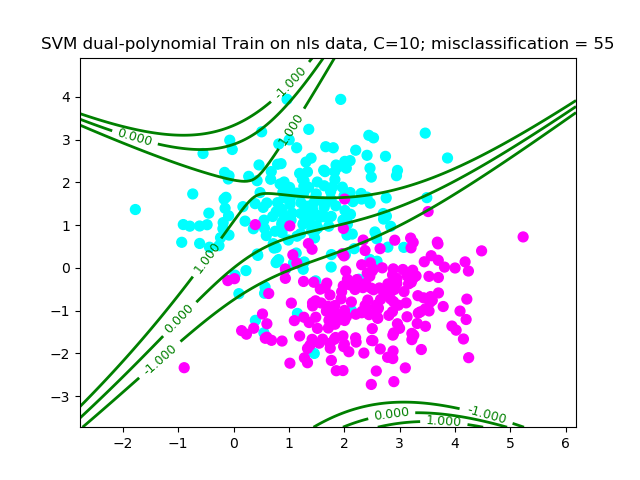
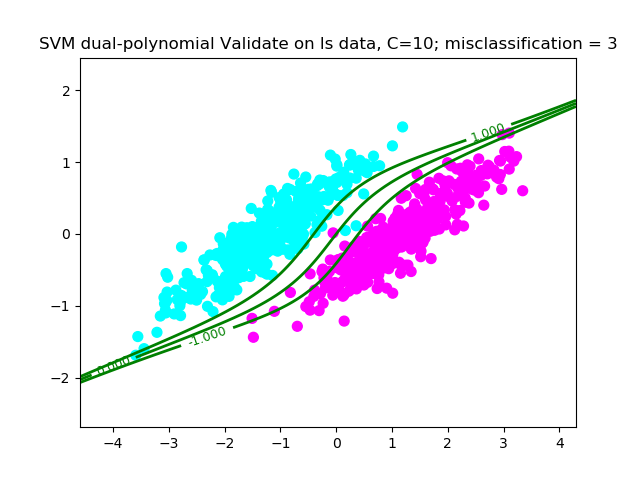
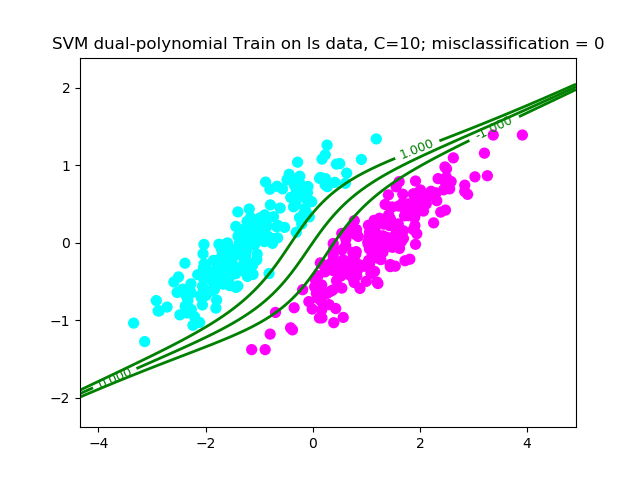
**Q2. Support vector machine implementation**

SVM Dual and Primal form for the data sets for various C

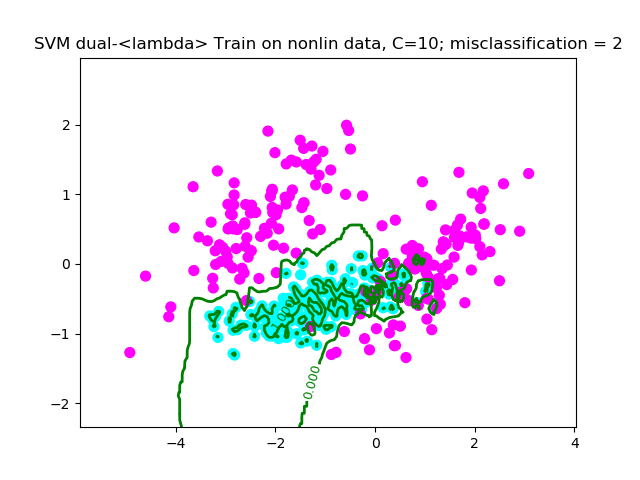
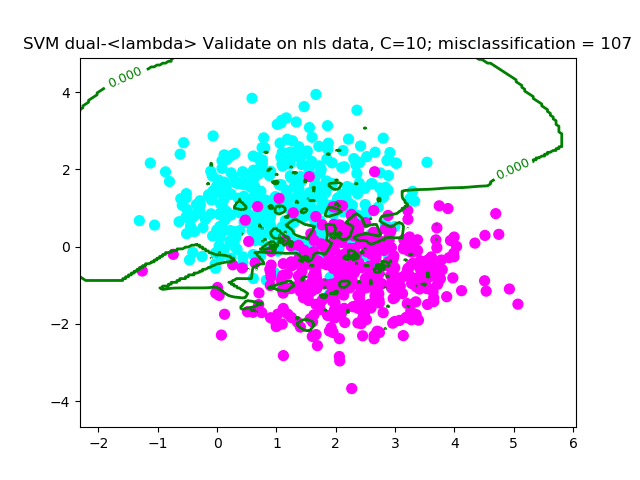
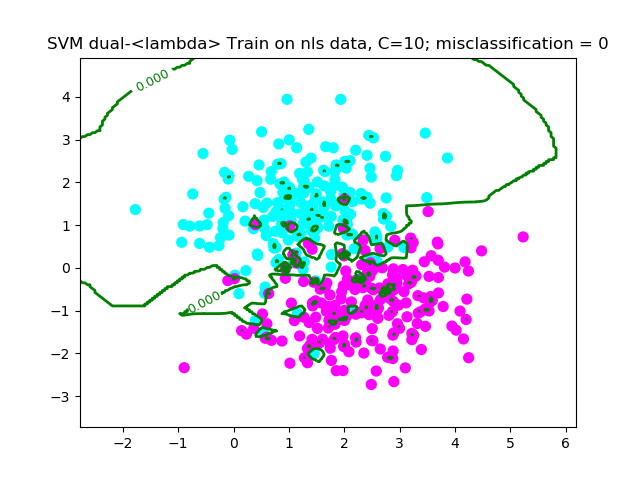
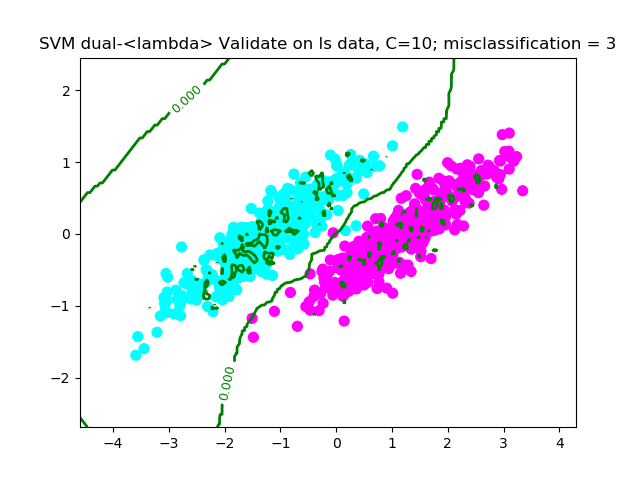
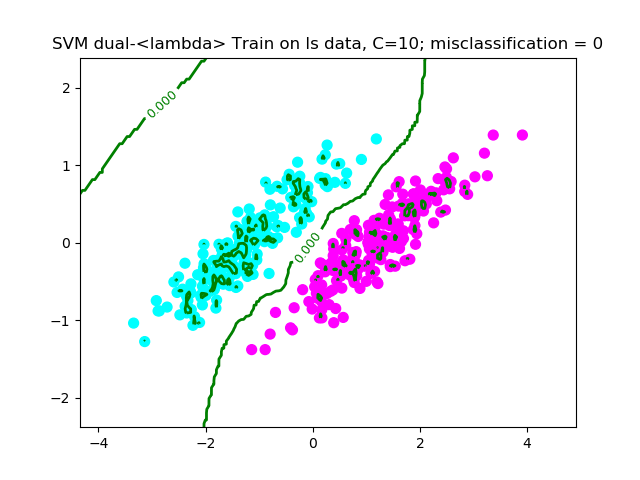


**Q3. Kernel SVM**

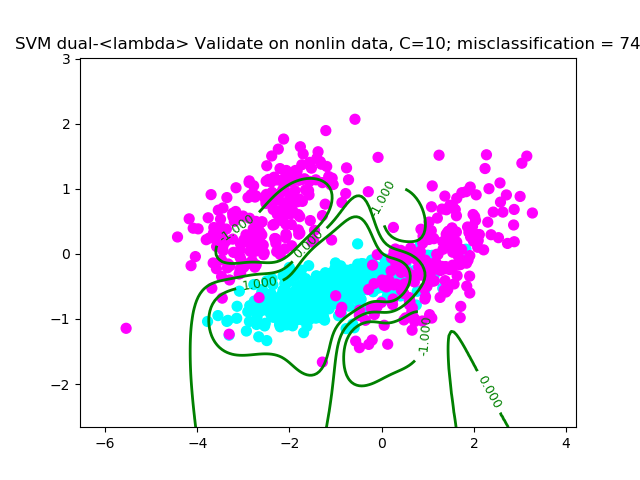
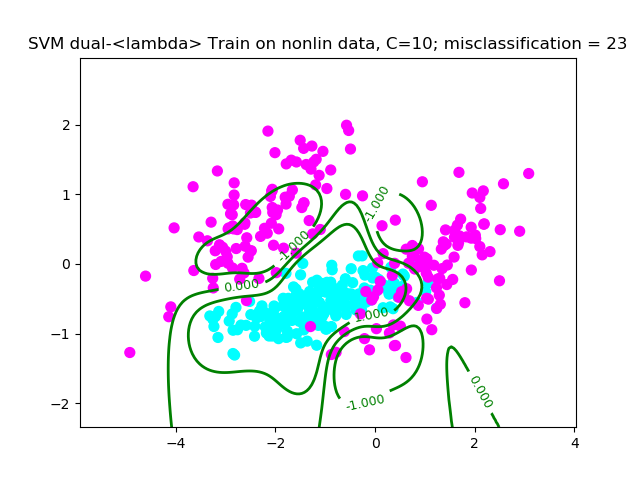
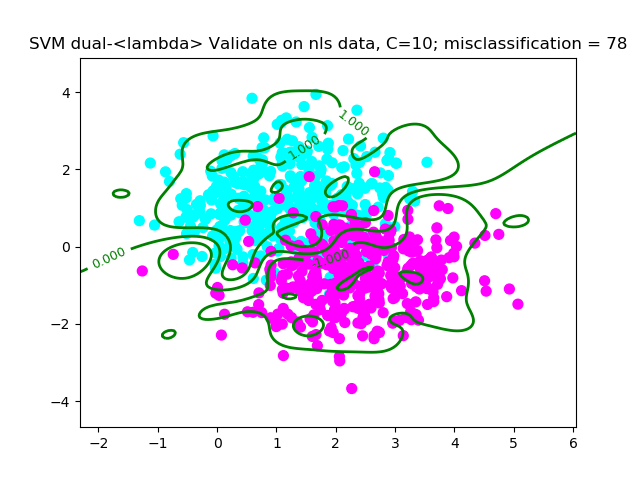
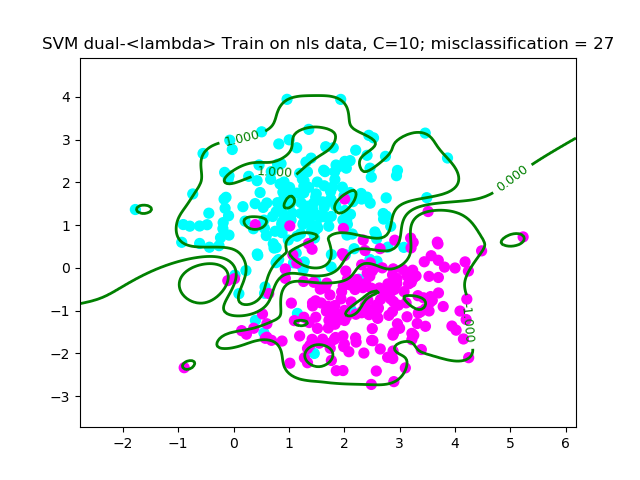
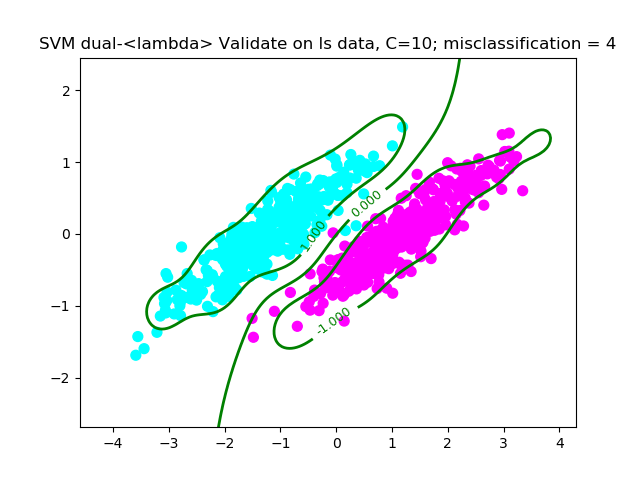
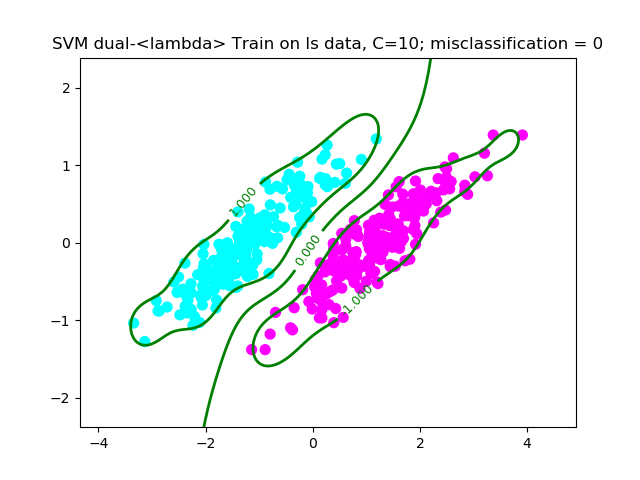
Polynomial kernel



Gaussian kernel with sigma = 0.05



Gaussian kernel with sigma = 0.5



Gaussian kernel with sigma = 5

