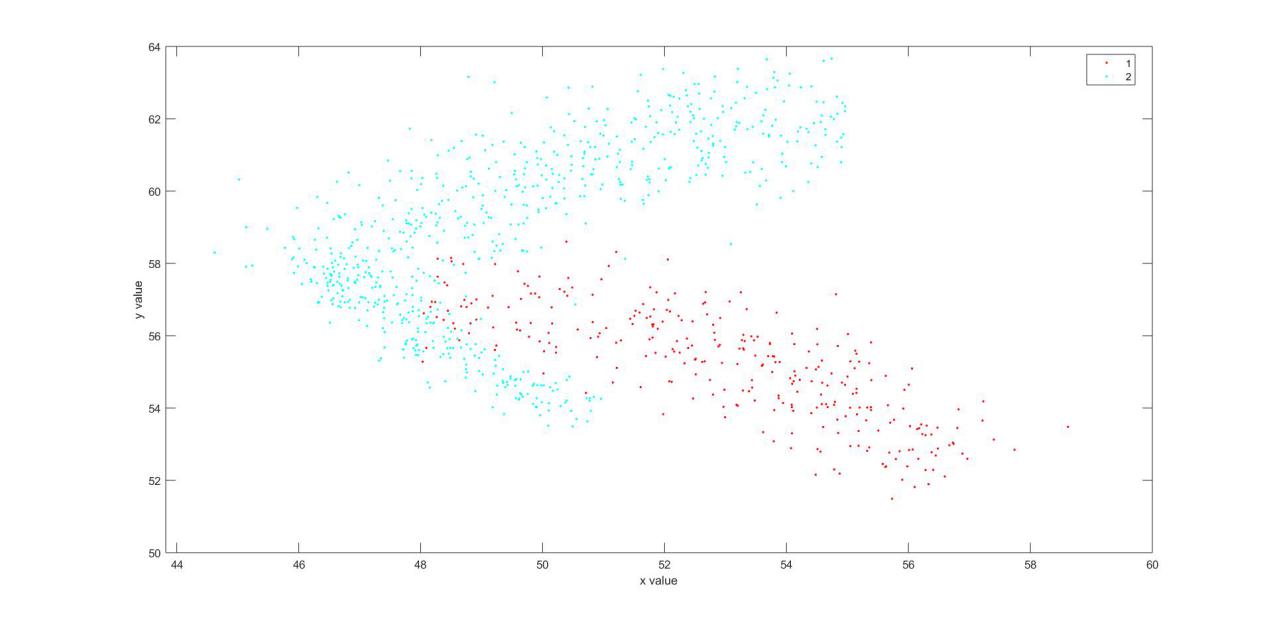
BIOINF 580 - Assignment 5

Due on April 21, 2019 at 10PM

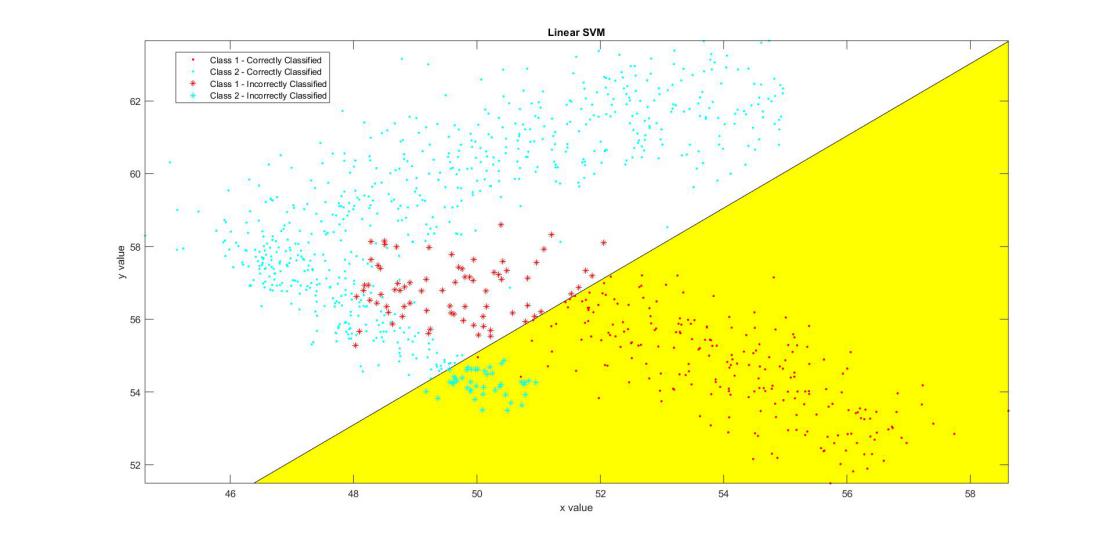
After implementing each step, copy the output, the resulting figure and the description of your results into this document. To copy a Matlab figure, click on *Copy Figure* in the edit menu of your figure. Make sure that your *x*-axis has the appropriate unit, all your axes are labeled and your plot has a title. Submit this document as part of your assignment along with your code(s). **Your codes should be submitted as separate Matlab (.m) files (not as part of this document).**

1. Download *data3.mat* from the assignment folder and open it in Matlab. Separate 20% of the data for testing and the remaining 80% for training and use these partitions for all parts of the question 1 (use *cvpartition()* function in Matlab and the *Holdout* option).
   1. Display the data in Matlab using different colors to distinguish between classes. Comment on the structure of the data. What kind of classifier/kernel do you think is most appropriate to classify this data? (3 points)



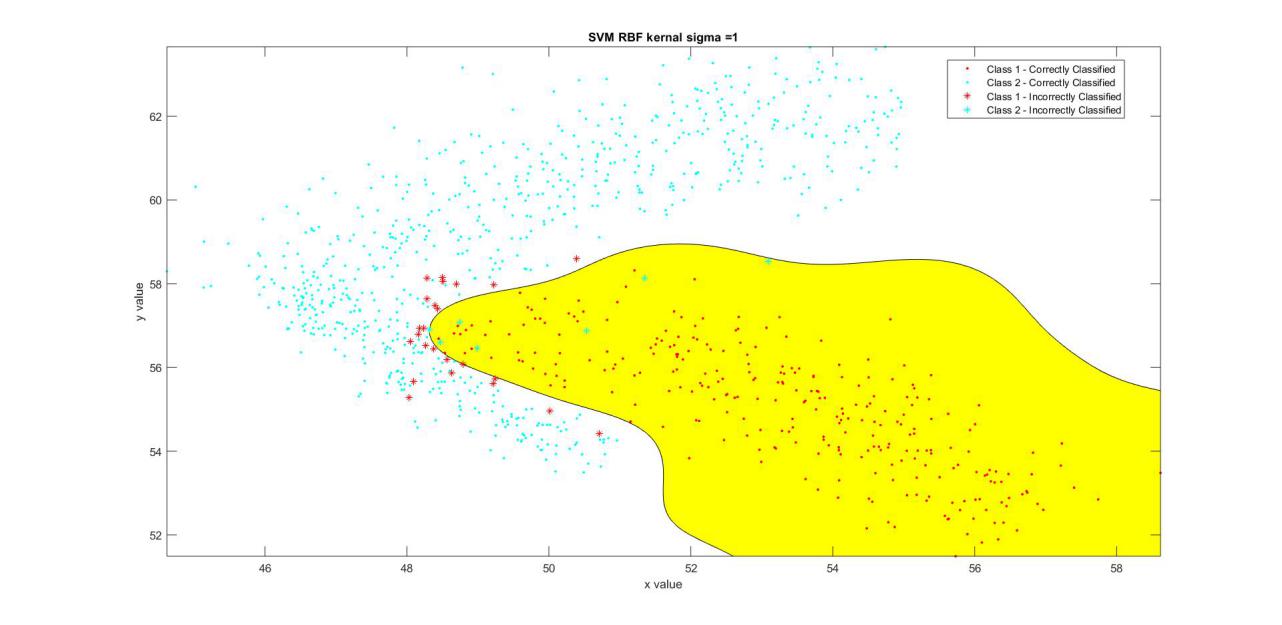
***As the boundary of the data is not linear, require some non-linear kernals. RBF kernal can be used to account for the non-linearity of the boundary.***

* 1. Use a linear SVM classifier with C=1 to model the data. Display the classes and the class boundaries using *svm\_plot\_classes()* function (available on Canvas as part of Lab 6 package). Comment on the results and the quality of the classification. (2 points)



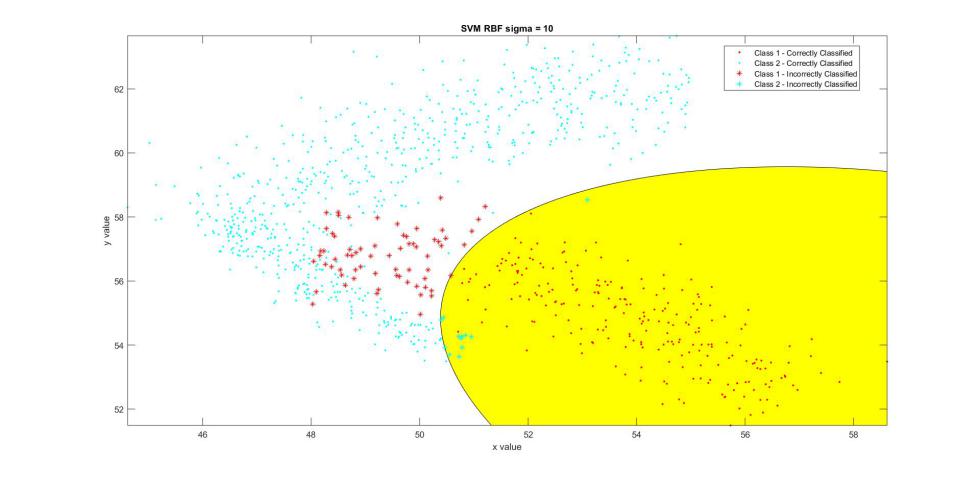
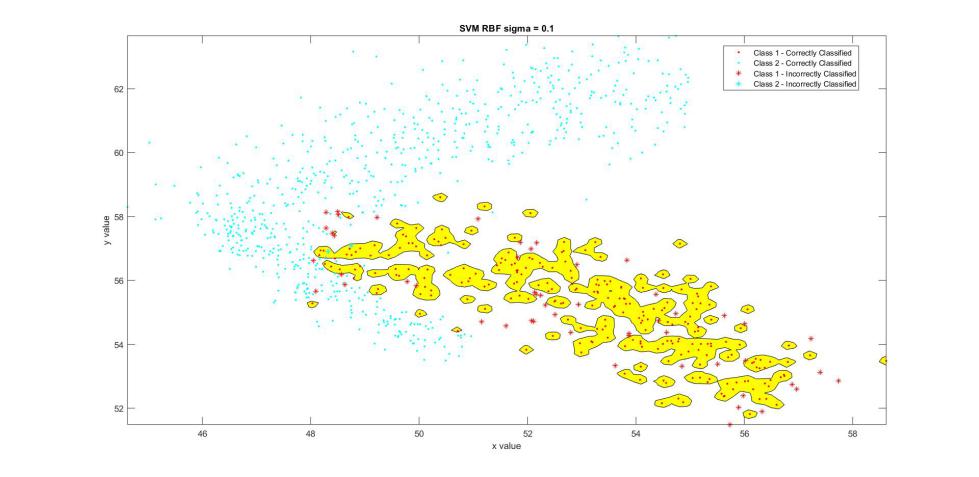
***The predictions is poor, for training dataset, the accuracy is 0.8840. For testing dataset, the accuracy is 0.8760.***

* 1. Repeat part b with an RBF kernel and sigma=1 for the RBF kernel parameter. Comment on the quality of the classifier and compare it to the linear kernel. (2 points)



***The quality is much better. The accuracy for training data is 0.9720. The accuracy for testing data is 0.9600***

* 1. Use sigma=0.1 and sigma=10 for the RBF kernel with C=1 and compare the results. Explain why the differences occur. (2 points)



***When sigma = 0.1, the model’s boundary has a very low reach for its data. it only goes around with value in very close proximity of the training data. Therefore, the models applied to training data could have accuracy 0.9933, but the accuracy for testing data is 0.8040, suggesting the model is overfitting the training data.***

***When the sigma is set to 10, as the boundary is determined by data that are further away, the models is more general. The accuracy for training data is 0.9133, and the accuracy for testing data is 0.9440. Certainly, this is not an overfitting of the data. However, as those data that is far away from the seperation boundary could have a lot of effect on the boundary, the boundary failed to seperate out the value that is in close proximity of the other groups.***

* 1. The C parameter for the SVM classifier controls the balance between the rate of misclassification and the complexity (larger minimum margin) of the model. The higher the value of C, the higher the complexity of the model and the lower the misclassification rate on the training set, and vice versa. Vary C parameter for the SVM classifier (with RBF kernel and sigma=1) from 0.1 to 1000 (on a log scale) and visually inspect the impact on the classification results. Comment on the results. What C parameter do you think leads to the best results? (4 points)

***As the C parameter increased, the accuracy for training data increased from 0.9600 to 0.9853. However, the accuracy for testing data is 0.9720,0.9760,0.9760,0.9720 and 0.9680 respectively for sigma value equals to 0.1, 1, 10, 100, 1000. I would not say this is an overfitting of the data as the boundary for seperation did seperate the data quite well. However, when C equals to 1000, there is a hole in the region that is mapped to class 1; additionally, the accuracy for testing data did went down, suggesting the model does not need to have a higher complexity to lower the misclassification rate on the training set, therefore it is a little bit of the overfitting the data. As C=1 has a lowest model complexity among all the most accurate results, I think C = 1 leads to the best results.***

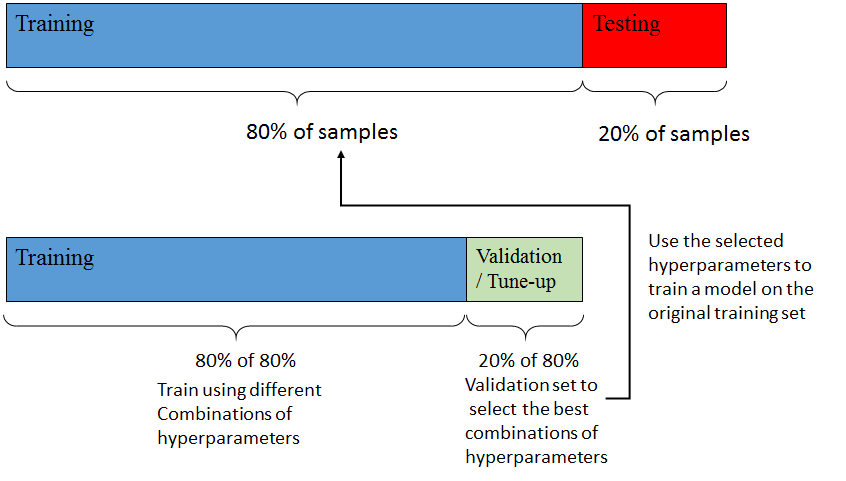
* 1. Repeat part b with a polynomial kernel and degree=5 for the polynomial kernel parameter. Comment on the quality of the classifier and compare it to the RBF kernel. Check Matlab’s documentation for fitcsvm function using polynomial kernel. (3 points)

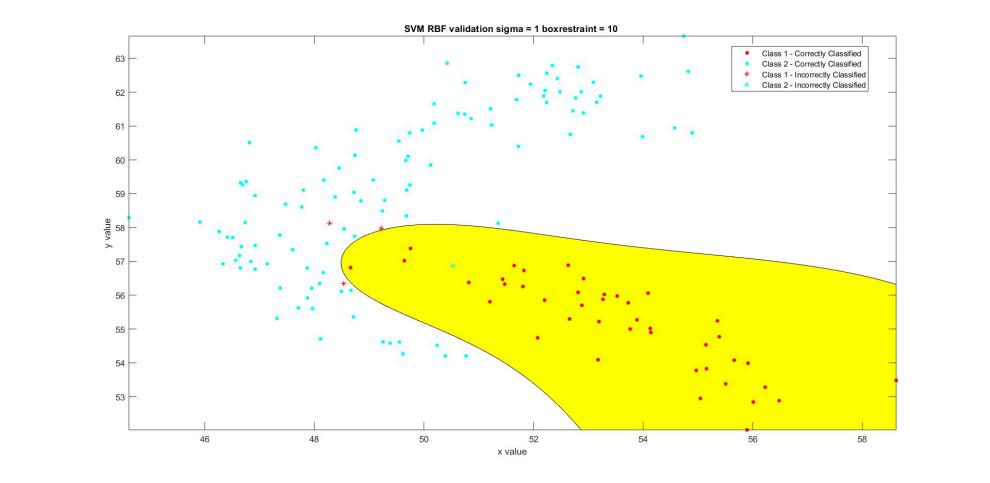
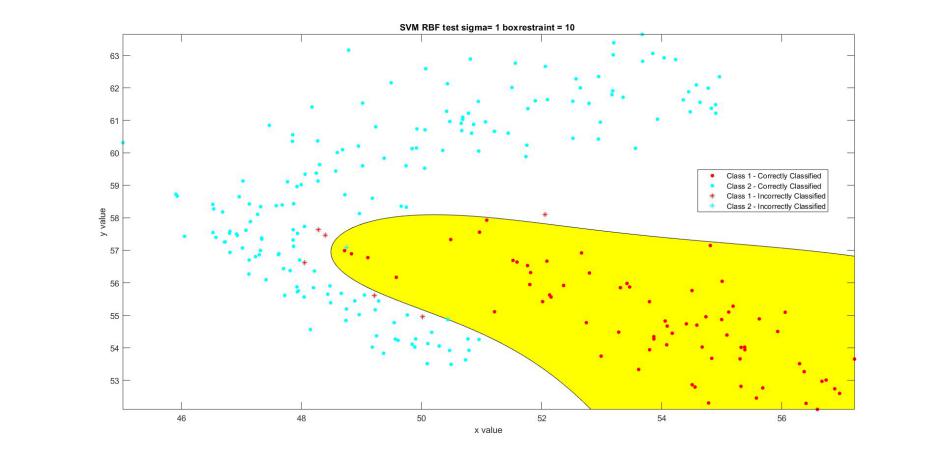
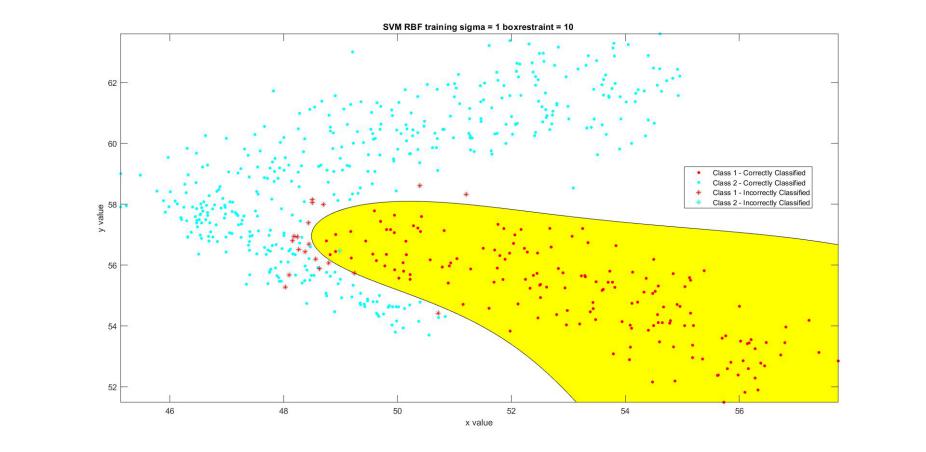
***The SVM with polynomial kernal at degree = 5 is unstable. The most degree that is stable is when degree equals 2 and degree equals 1. When degrees equals 1, the test accuracy is 0.9160 and when degrees equals to 0.7960. When C = 1, I believe the polynomial kernals are having a lower accuracy comparing to RBF kernal. Even it might theoretically have a better accuracy, but unstable results indicating the polynomial kernel in this case is not well-performed.***

* 1. Use degree=2 and degree=10 for the polynomial kernel with C=1 and compare the results. Explain why the differences occur. (2 points)

***A range of degree from 1 to 10 is tried. Only degree 1 and degree 2 for the polynomial kernel is stable. Though the plot cannot be calculated, but from algebra, as the polynomial’s degree getting higher, it will have a higher freedom to better fit the curvature of the boundary, thus having a better accuracy.***

* 1. Separate the training set into 20% for validation and 80% for training (below structure). Perform a grid search to Iterate over sigma and C values (you can use 0.1, 1, 5, 10, 50, 100, 500 and 1000 for both parameters) and use each combination to train an SVM model on the training data. Compute the accuracy of the trained model on the validation data and store the accuracies in a matrix. Then, find the best combination of parameters that leads to the maximum accuracy on the validation data and use it to train a model on the training and validation sets combined. Apply the resulting model to the testing data and plot the outputs using the training, validation and testing sets **(three separate plots).** Comment on the results. (6 points)





***It has best performance when sigma equals 5 and box restraints equals 500. For training data set, the accuracy is 0.9667; for validation data set, the accuracy is 0.9733; for testing data set, the accuracy is 0.9720. This is not a overfitting results as the accuracy for validation and testing is higher than that of the training data.***

* 1. Compute the true positive, true negative, false positive and false negative rates as well accuracy, F1 score, recall and precision for the training, validation and testing datasets using the tuned hyperparameters. Interpret the results. Which results do you think represent the true performance of the classifier? (2 points)

***All the rate is stored in matlab. I believe all of the rates and score could represent part of the performance of the classifier. In order to represent the true performance, I believe, the results should only accounts for the testing datasets and validation datasets. F1 score, which is a function of recall and precision, better represent the performance. Additionally, accuracy would always represent part of the performance.***

***For validation data set, the accuracy is 0.9733, F1 score is 0.9545; for testing data set, the accuracy is 0.9720, F1 score is 0.9517.***

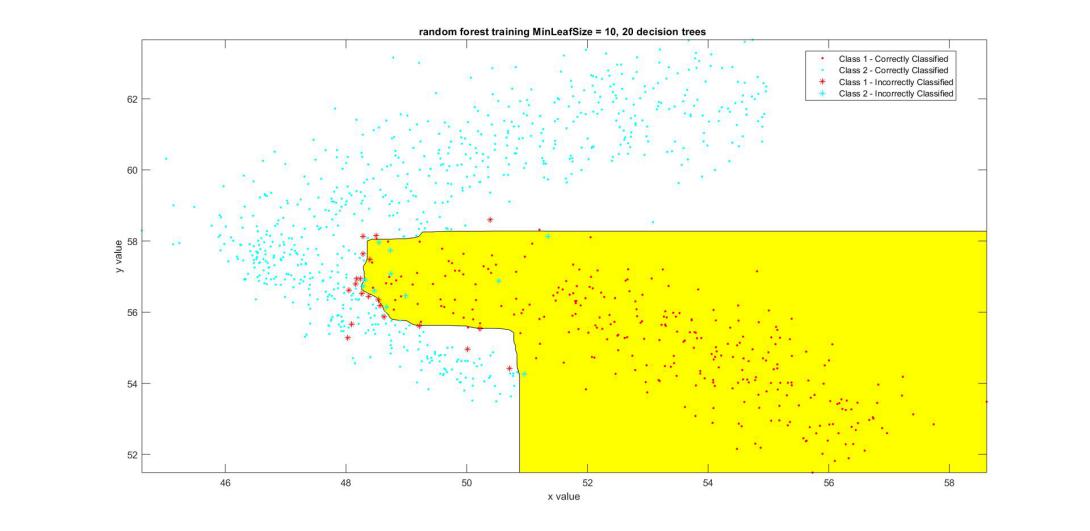
**\* For question 2 to 5 use 10-fold cross-validation.**

1. Use sigma = 5 and C = 5 for the RBF kernel to train SVM model. Compute F1 score and AUC of ten trained models (one for each validation fold) on their corresponding validation folds. You can use Maltab’s perfcurve function to calculate AUCs, but keep Dr. Najarian’s comment on why it is not the best practice to calculate AUC in mind. (4 points)

***F1 score is : 0.9286,0.9091,0.9153,0.9474,0.9831,0.9474,0.8727,0.9286,0.9474,0.9474***

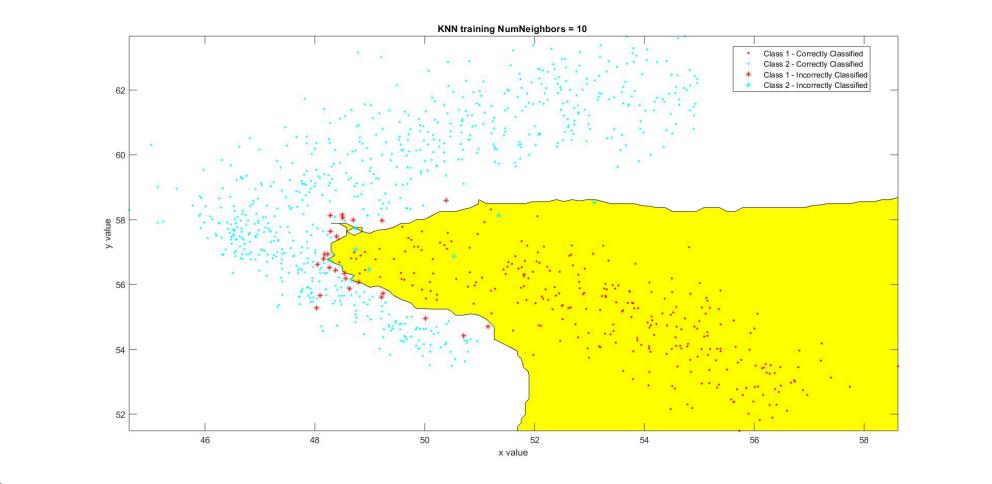
1. Train a random forest classifier with 20 decision trees, and set the minimum leaf size to 10. Display the classes and the class boundaries using *tree\_plot\_classes()* function available in lab\_6 package. Compute AUC of ten trained models on their corresponding validation folds. (4 points)

***One of the plot is shown here.***



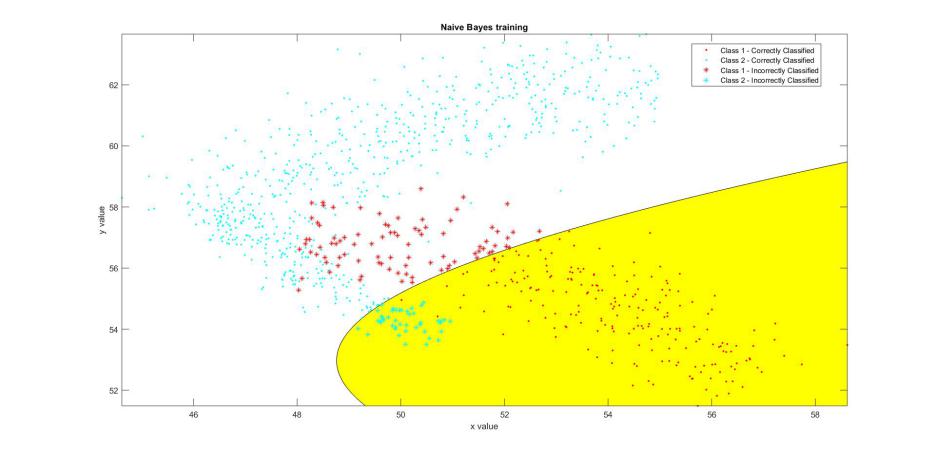
1. Train a k-nearest neighbor (KNN) classifier by setting the number of nearest neighbors to 10. Compute AUC of ten trained models on their corresponding validation folds. Display the classes and the class boundaries using *nb\_and\_knn\_plot\_classes()* function available in the assignment package. (4 points)

***One of the fold’s plot is shown here.***

******

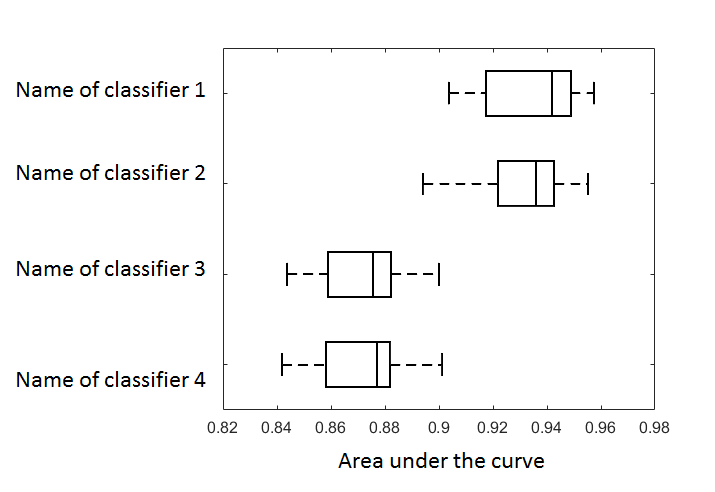
1. Train a naïve Bayes classifier. Compute F1 score and AUC of ten trained models on their corresponding validation folds. Display the classes and the class boundaries using *nb\_and\_knn\_plot\_classes()* function available in the assignment package. (4 points)

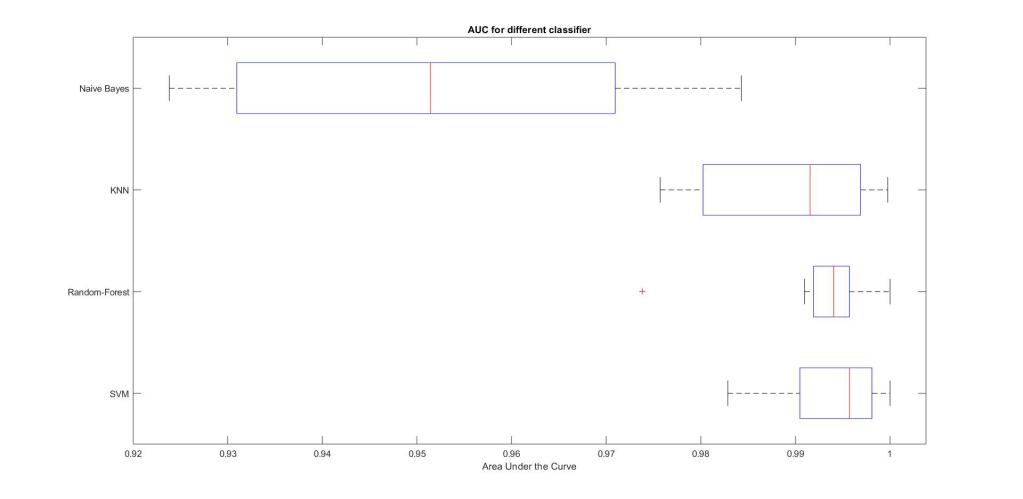
***One of the fold’s plot is shown here.***

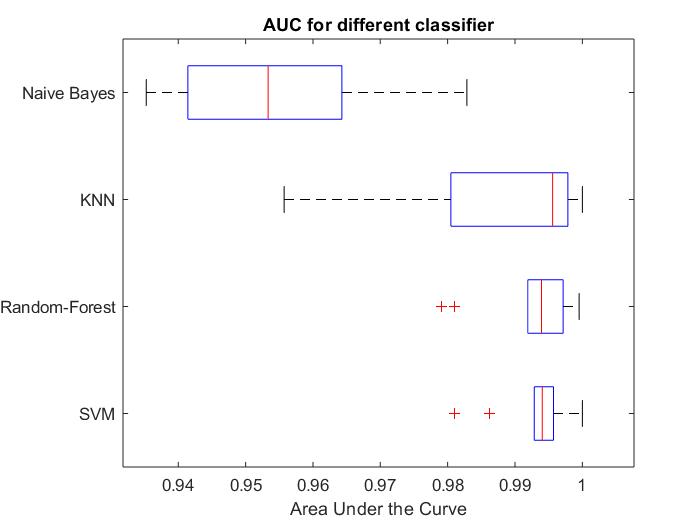
******

***F\_score: 0.8667 0.6923 0.6531 0.7636 0.8000 0.7458 0.8475 0.7667 0.6786 0.7778***

1. Generate **one** figure containing 4 boxplots, one for each classifier of questions 2 to 5, using computed AUCs over 10 folds (check the below example plot for the correct format). Compare the generated plots and comment on the model you would choose for this problem. Why? (8 points)







***Due to the wrong doing of confuse with the sigma value with the kernel scale value, the replot of the AUC give’s me a different answers. The first plot is the plot where I ran the classifier for the first time with kernel scale value equals 5. The second plot is the plot where I reran the classifer to change the kernel scale value to 5\*sqrt(2). This wrongdoing actually provides insights of this problems. As the parameter for each classifier is not optimized, there is noway to conclude which classifier is better based on this AUC curve. For the first plot, Random\_Forest has the best AUC performance while for the second plot, SVM has the best performance, suggesting, the unoptimized parameter has a huge effects on the performance. However, from visual discrimination, Random-Forest’s seperation boundary is more linear than KNN, SVM and Naives-Bayes, which did not fully keep the data’s originial structure. Naives-Bayes did not perform well on the data so I’ll will pick between SVM and KNN. The mean of the F\_score of SVM is smaller than that of KNN, therefore I will probably pick SVM as the classifier for this problem if I have to pick one.***

Turn in the following files:

* *main\_function\_5.m*
* This report named youruniquename\_hw5.dox