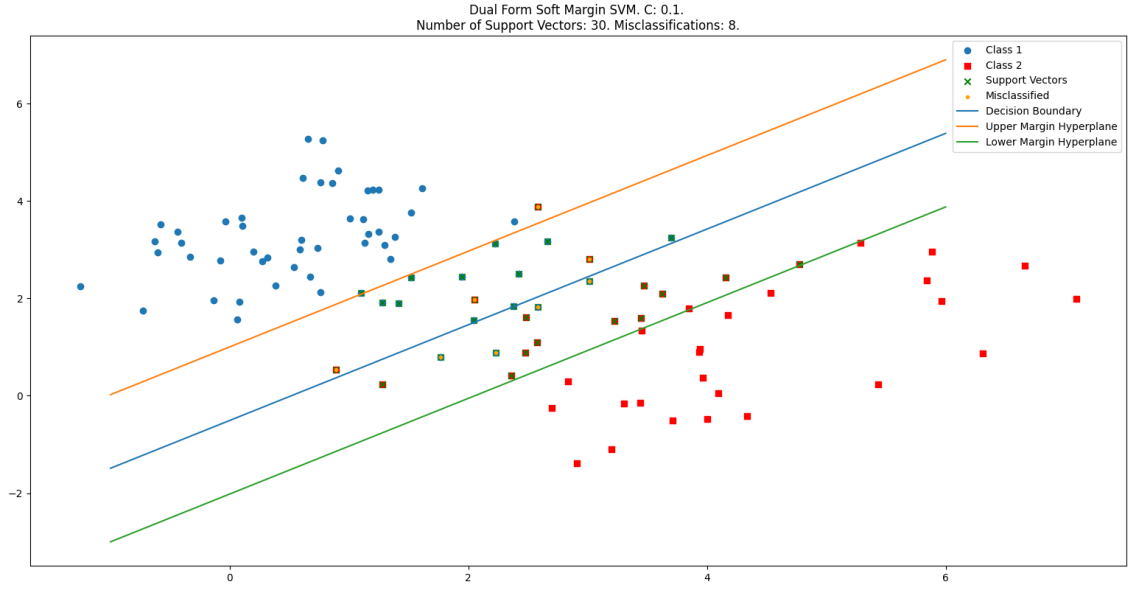
Pattern Recognition

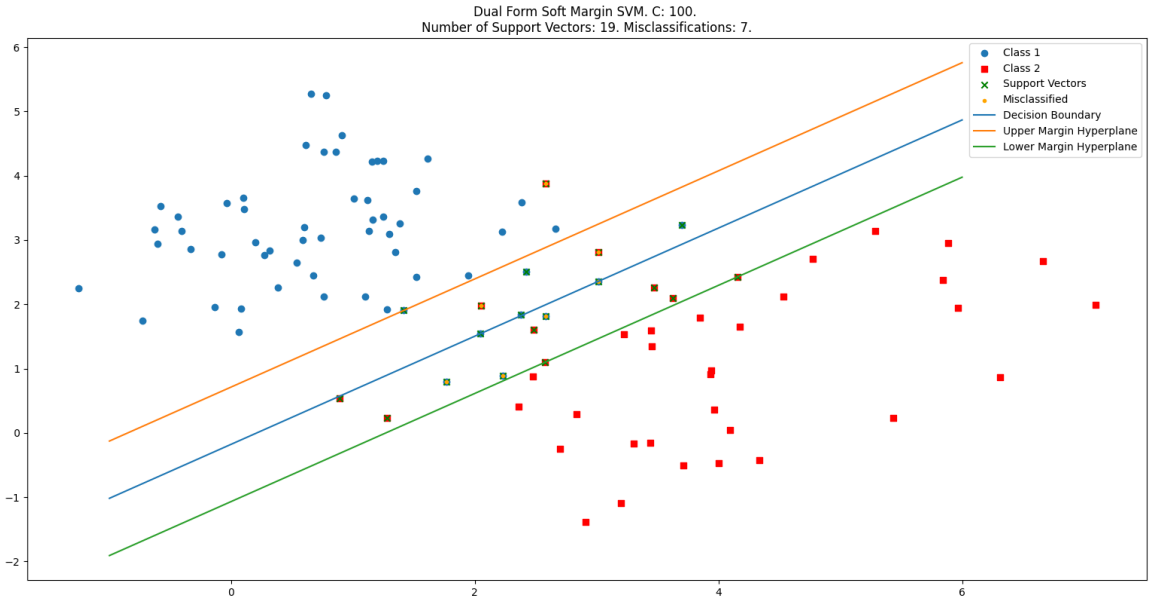
ECE 4363 / ECE 5363

Project 2

1. Use Matlab's quadprog() function or Python's CVXOPT package to implement the linearly nonseparable (soft margin) SVM in its dual form and test its functionality with the data set generated as shown below. For $C = 0.1$ and $C = 100$, plot the samples, margin hyperplanes, and the decision boundary. Also, on the plot, identify and give the count of the support vectors and the misclassified samples.

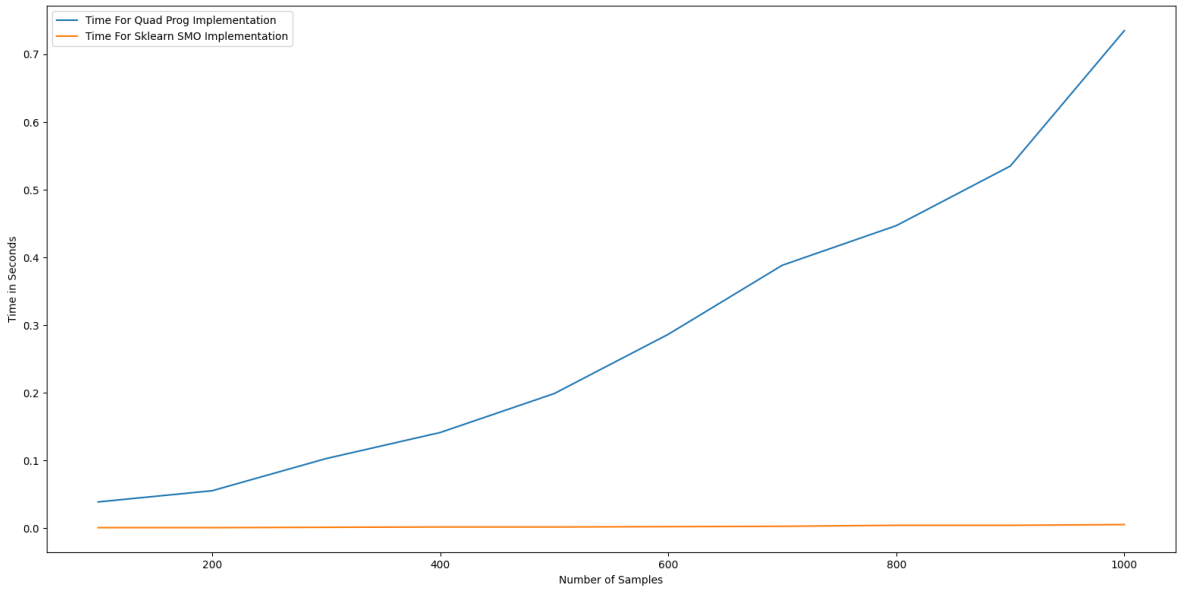


First plot. C = 0.1. 30 Support Vectors. 8 Misclassifications. 1 Misclassification that is outside of the margins (the top red data point).



Second plot. C = 100. 30 Support Vectors. 8 Misclassifications. Thinner margin, 1 less misclassification. Visually looks like it's probably a better fit actually despite higher C usually leading towards higher overfitting.

2. Compare the computational efficiency of your implementation of SVM with one in Matlab or Python that employs the SMO approach. Present this comparison as a plot of the number of training samples versus execution time.



Plot of number of samples versus time to execute each method in seconds. I did 10 iterations at 100,200,300,…,1000 samples each. The quad prog implementation grows almost exponentially. The

other method I used is Scikit-Learn's SVM package that uses the SMO algorithm. It looks like a flat line on this scale but it's actually growing almost linearly. Much faster than my implementation (by a couple orders of magnitude).