1. (1 point) The primal problem for Support Vector Classification is expressed as follows:

$$\min_w \frac{1}{2} w^T w \quad \text{s.t.} \quad \gamma_i (w^T x_i + b) \geq 1 \quad i = 1 \ldots n$$

Why is the minimization of $w^T w$ so crucial? What is the geometric interpretation behind it?

2. (2 points) In the soft-margin SV Classification primal problem, the $c$ parameter plays a crucial role.

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_i \xi_i \quad \text{s.t.} \quad \gamma_i (w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0$$

(i) Can you describe the role of $c$ in your own words? What happens as $c$ gets larger? What happens if it is set to zero?
(ii) What is the proper name used for the $\xi_i$ variables? What happens if $\xi_i > 1$ and why?

3. (2 points) In our SVR discussions, we talked about an $\epsilon$-tube.
(i) What happens if $\epsilon$ is set too large?
(ii) There is a soft $\epsilon$-tube approach described in one of the recommended papers. Can you tell me a little bit about this?

4. (2 points) Write down the Lagrangian function for the primal SVC problem in question (1).
(a) Take the derivative of the Lagrangian function with respect to $b$.
(b) Take the derivative of the Lagrangian function with respect to $w$.
(c) Plug in the derivative of (b) into the Lagrangian function and derive the dual formulation of the problem.

5. **Bonus Question** This question is worth 3 points and I will also award an extra 1 point and a beer for whoever can match or beat the best performances thus far. (proof of your work will be required for verification)

The data: I have placed some protein data out on the course website. This is real data that I have worked on myself. This data pertains a set of 220 proteins that are described by a set of 34 features. The proteins fall into two classes labeled as 1 and 2. The data is randomly partitioned into 10 train/test sets and you will find these under the zipped files train and test. The protein labels are in the very last column of each of these files.

Research thus far: Using an RBF kernel and SVMlight, and doing a grid
search on the $c$ and $\sigma$ parameters, the smallest misclassification error I achieved was 18 out of 220. One of my collaborators has also worked on this data and, using Random Forests and PCA, his misclassification error was also 19 out of 220. **Can YOU do better?**

Your task:
Decide what numerical kernel method you are going to use. If you are going to use RBF (Gaussian), then you may want to consider how to choose the $\sigma$ parameter. You may also want to consider things like normalization and kernel centering.

Choose an appropriate supervised classification method, for example, Support Vector Classification - Weka is a good collection of tools that you may want to consider.

Set your kernel and learning parameters. For each train/test set, implement the learning algorithm and then test it out on your test set. Keep a count of the misclassified points as you progress from one train/test set to another.

Your report: With your optimal settings, I need to know:

- How many test points were misclassified?
- What kernel method did you use? what kernel parameters (if any)?
- What learning algorithm did you use? what software? what were your parameters?