Here are some guidelines for writing your project reports. In the past, project reports have been from 8 to over 20 pages long, with most in the 12 to 15 page range (not counting appendices, which can be a bit longer). In the past, most projects have been individual efforts, so group reports might tend to be a little longer. Please do not turn in large sections of code listings or massive tables of raw data (although some information on the data is important, and if you have an unusual data set a table indicating what typical examples look like would be OK). The report should be easy to read, if I can’t tell what you are talking about it is difficult to give you a good grade. Please typeset the report in 12 point type (the 10 point default used by LaTeX is too small, this document is in 12 point). Every figure or table in the report body should be discussed in the report body. If you would like to present additional experiments that are not evaluated in the body of your report, include them as an appendix.

You should define what the symbols you use mean when you first use them, and be consistent. Try not to overload the same symbol with more than one meaning (don’t use \( n \) for the number of examples in the sample and then later write \( \sum_n \)). If you use a lot of symbols so they may be hard to keep track of, include a table of notation where each symbol is listed together with a short description of what it represents (see pages xxix and xxx of Alpaydin for an example).

Your report should have an abstract as well as introduction/problem description, related work, methods used, results, and conclusion sections, as well as a bibliography. I am flexible on the exact section breakdown, add or merge sections if it makes writing/reading the report easier. Readability is important, so be sure to define your terms before using them and present things in a logical order.

Your report should start with a short 1-paragraph abstract that mentions the problem you attacked, your main methodology, and your results (perhaps 3-4 sentences total). You can use standard technical terms without definition in the abstract.

The introduction should provide an overview of what the problem is, why it is interesting/important (why did you choose it) how you attacked the problem, and an idea of your success and/or failure of your methods. The description of your problem and whatever background is needed should be at the level that any student in CMPS 242 can easily understand. The introduction also gives an overview of your main results, and can include information on how you obtained your data, etc. If this additional information also appears elsewhere in your report, then it should be summarized in the introduction (perhaps with a forward pointed to where it is discussed in more detail). If your particular problem is too technical or difficult to describe precisely in the introduction, then you might give just an overview in the introduction and use a different section to describe it precisely. The introduction can also contain the specific terminology related to your application. (A word or phrase’s is usually set in italics where it is defined, like: “An example is an instance-label pair.”) Definitions can appear in other sections also, so you don’t have to force all your definitions into the introduction. If you have a lot of jargon (terms specific to your methods and/or application) feel free to include a a table of definitions.
The related work section should contain a survey of relevant previous work for your problem and possibly the methods you used. This is sometimes a good place to clearly spell out what you did for the course as opposed to what was done by others or outside of the course. Feel free to cite textbooks or articles etc. for descriptions of algorithms. However, the best related work sections are not just lists of references, but evaluate, compare, and put into context the previous contributions, as well as relating them to the current work.

The methodology section should describe the details of your experiments. This includes the data source (what was measured, how, what kinds of errors are likely) and any pre-processing done. Describe the learning techniques used and what software packages (Weka, SVM light) you used. Ideally, there will be enough information here so that another student could reproduce your results. Although I am not interested in a printout of any code you wrote, you could include a link or pointer to where it could be obtained (as well as your datasets). You should also explain here (if not earlier in your report) why you picked the methods you did.

The experimental results section should describe what happened. Is it what was expected? What were the surprises/anomalies? In retrospect, why do you think the results come out the way they did? How do your results compare with others? Ideally, each experiment is a question and the results provide an answer. Tables and graphs are appropriate ways to summarize information. If you are doing many experiments or varying many parameters, a good way of structuring your presentation is to have a baseline or default set of parameters and compare each of the individual experiments to the baseline. If the focus of your project is how a complicated algorithm works on a problem, a comparison with the performance of a simple algorithm is a good idea (like Naive Bayes may be a good candidate for this depending on the particular application).

The conclusions section should include a short self-evaluation of your project (what went right and what went wrong) together with a summary of what was learned from the experiments and what you yourself learned from the process, as well as a recap of what you accomplished. If there are other things you would have liked to try but didn’t get around to, you can include future work in the conclusions section (or even make further work it its own section).

You should acknowledge any help you have been given on the project and anything else that made the project possible (such as data or machinery/code that you used).

The bibliography should contain relevant publications (articles, books, etc.) that you read in conjunction with your project. It is bad practice to cite something without at least having skimmed it.