CS303 Project Report Grading Policy

Structure

A report should at least include the following sections. The order given here is suggested but not necessary. You can re-organize it as long as the expected information is presented.

1. Introduction (15 points)

1. Give a general introduction to the problem studied in this project. For example, where does it originate from, how can it be characterized, and to what kind of real-world problems can it be applied?
2. State the purpose of this project/report.

2. Preliminary (15 points)

Formally formulate the problem, and explain the terminology and notation you will use throughout this report.

- A formulation is an abstract but accurate description of the problem. It should disambiguate the potential confusion in natural languages.
- Example: The problem is a supervised binary classification: given a feature vector x (demographics and census attributes) predict label $y \in \{0,1\}$ indicating income $\leq$50K (0) or >50K (1); the goal is to learn a classifier $f_\theta(x)$ that maximizes prediction accuracy (or equivalently minimizes empirical risk).

3. Methodology (30 points)

1. General workflow (5 points).
    - Example: The proposed method divides into steps 1, 2, and 3, each involving algorithms A, B, and C, respectively.
2. Detailed algorithm/model design (20 points).
    - Describe the essential part of your algorithm/model with pseudo-code/flow charts/diagrams.
    - DO NOT paste (edited) Python code.
    - If there is some complex data structure that is not intuitive to understand how it is implemented, give additional explanations.
3. Analysis (5 points). Discuss, for example, the optimality and complexity of your algorithm, and what is the deciding factor of its performance.

4. Experiments (30 points)

1. Setup (8 points).

- Give a short introduction to the dataset(s) you used (if any). Try to comment on the dataset's characteristics and relate them to your analysis.
- Give a description of the environment, e.g., software/hardware configuration, Python and NumPy versions.

2. Results (12 points).

- State the performance measures of interest, e.g., running time, optimality, classification accuracy, regression error.
- Experimental results (using tables, figures, charts, etc.)
- Try to find through the experiments:
    - the effect of different optional components of your algorithm (if any),
    - the effect of hyperparameters (if any).

3. Analysis (10 points).

- Comment on the experimental results. Are they good or bad? Do they meet your expectation/hypothesis?
- Analyze the effect of different components and hyperparameters if you have corresponding experiments.
- Try to relate the results to the Methodology part. For example, discuss the relationship between theoretical time complexity and the actual running time, give possible explanations for the discrepancy (if any).

5. Conclusion (10 points)

Draw informative conclusions from what you have done and written.

Possible things you can write:

- Comments on the advantage/disadvantage of the algorithm you used.
- Does the experimental result match our expectation/analysis.
- The lesson you learned from this project. E.g., how to get a fast implementation in Python?
- Further thoughts on how it can be improved.
- ...

Writing

Additionally, you will be graded on the correctness and clarity of your writing.

Here are examples that you will need to notice:

1. Be concise. The suggested length is 3 to 5 pages. You will get -3 for writing too long.
2. Be careful. You will get -3 for poor formatting or too many grammar mistakes.
3. List the references in a valid format. Also, indicate the source if you have used something from, e.g., GitHub or CSDN.

Tips

1. Use Overleaf or ShareLaTeX provided by SUSTech CRA to write reports/articles in LaTeX.
2. Getting a IEEE article template:

   A. Go to the template selector page.

   B. Make some selections and download the template as a .zip file.

   C. Log into Overleaf or ShareLaTeX, choose New Project -> Upload Project and upload the downloaded .zip file.

3. Pseudo-code in LaTeX.