EEE549: Statistical Machine Learning

Final Project Report Grading Rubric and Expectations

This rubric is LONG. Read it carefully. **The first part explains the expected effort of the project and the second part explains how the results should be organized.**

Here is an overview of what the project is about and what is expected from your work.

- **Datasets**:
    - Tabular: UCI Adult and Wisconsin Breast Cancer Datasets
    - Numerical: Fashion MNIST

- **ML Algorithms (these are really called models/hypothesis class)**: **NO CLUSTERING**
    1. Logistic Regression with and without regularization
    2. SVM with and without kernels
    3. PCA+k-NN (because PCA by itself is not an inference algorithm!)
    4. FNN (for tabular data) or CNN (for image data)

- **Results expectation**:
    - Test every model using a principled training, validation and evaluation procedure
    - Make this procedure clear upfront and detail specific differences for each algorithm
    - List all hyperparameters and which were tuned and which weren't – list tuning approach
    - For NNs, hyperparameters are generally done by random search exploiting knowledge of parameter space – it can become intense – better to be honest and clarify what is possible and what isn't and why the choice of the fixed hyperparameter values were made as so
    - If lack of computing power limits tuning learning rate etc., be specific and be precise about the choice of values for ALL hyperparameters involved. For example, SVM-kernel method has a set of hyperparameters that the function called uses or you input to it – all these should be listed, which were tuned should be listed, and which were perhaps fixed (perhaps as an example, learning rate?) and why at that value should be listed.

- **Report length**: 8 pages + 1 page extra for references
    - these are recommendations -- you can always write a shorter but NOT a longer report
    - Report should contain details on each algorithm
        - library/libraries used
        - Hyperparameters listed

    - Values or range of values for them
    - Validation approach for hyperparameter tuning
    - Plots of results
    - Metrics for performance
    - Dataset details upfront
    - Comparison for any algorithm across hyperparameters to highlight choice
    - Comparison across algorithms to highlight the effect of algorithm choice and the corresponding complexity of model and CONCLUDE (among the four choices) what works best for each dataset
    - All details on the architecture for FNN/CNN should be clarified and how it is changed to evaluate sufficient model complexity should be listed

- **Preferred document style**: one of the two below: **SINGLE COLUMN FORMAT**
  - IEEE/NeurIPS/JMLR style format (all these sample tex files are available online) <u>**(OR)**</u>
  - Word document as long as you either write equations with Word or use a high-def. copy-paste from relevant texts – all equations should be referenced to either lecture slides, textbooks, or papers as relevant

- **Main document**: <u>**8 pages**</u>; <u>**Section organization:**</u> Please use the same section titles as listed below from (a) to (j) below. Use below as an **approximate** guideline to write and stay within page limits but use your judgement as needed to explain.
  - **Names + Team contribution+ Brief project Introduction**: <= 1 page
  - **Section II**: 2 pages (or thereabouts)
  - **Section III**: 2 pages (or thereabouts)
  - **Section III**: 2 pages (or thereabouts)
  - **Conclusions, Value of course,**… : <= 1 page

- **References**: **1 page (beyond main document's 8 pages)**
  - **It is crucial to reference** Python library/function choice or mathematical details for every algorithm
  - All listed references must be <u>REFERENCED in the main document</u>.
  - You can either use the [number] system of IEEE or (Sankar et al.) style of NeurIPS
  - **All references should be hyperlinked**

- **Appendix**: for more detailed explanations and additional plots (**NO MORE than 6 pages**) ; Code can be extra
  - Necessary to include code (code will also be separately uploaded to Canvas and perhaps GitHub)
  - Additional plots can be added here but should be referenced in the main document

- **Code**:
  - Appendix of the report should contain the code. Add section titles for each ALGORITHM **AND** dataset
  - Upload as a separate file (**via canvas**)
  - ALL Details on how to run the code for each algorithm must be included in the Appendix; points will be deducted if not so
  - **If you use GitHub as a team for your code, include a clickable link in the appendix, reference in the appropriate places in the main document and make it public (do not ask grader for their GitHub ID)**

HERE IS HOW THE REPORT SHOULD BE WRITTEN

Main document details: [Total points: 75]

a) [**2 points**] **Title** of project and the EEE598 student names and programs on project

b) [**3 points**] **Making sure all students are listed in the final submitted report. Failure to do so will lead to a score of 0 for all students in the group.**

c) [**15 points**] (**Section I**): **Introduction**: (each bullet below should be a **subsection**) (subsections are numbered in Latex or in Word often as x.y where x is the section # and y is the subsection #; make sure you figure out how to do so)
   a. **[5 points]** Subsection of Section I: This is where you will **briefly** describe the datasets (completely detailed description, if desired, can be in the appendix prior to code for each dataset)
   b. **[5 points]** Subsection of Section I: Briefly describe the model classes and how they enable classification
   c. **[5 points]** Subsection of Section I: **Team contribution**:
      i. Make very clear what the contribution of each member of the team has been over the project and for the final report.

d) **[20 points] (Section II): Results for dataset 1**
   a. **Every subsection under this section will be one of the models**
   b. **Expect clarity on methods (see the first part of this rubric to clarify how to present)**
   c. Every mathematical term introduced should be defined precisely
   d. All figures should be captioned, referenced in the text, have clear axes, legends etc.
   e. **Code**: All code for each dataset should be in the Appendix – separate section for each dataset in the appendix.
      i. Use subsections for each algorithm for each dataset to make it easy to find the corresponding code
      ii. These section#.subsection# in the Appendix should be cross-referenced in the main document in the appropriate dataset+algorithm section for ease of search

f. **ALL CODE WILL BE TESTED.** **<u>Details on how to run it should be in the Appendix</u>.**
g. **If code is shared also via a GitHub link, then the code must be made public and a clickable link must be provided in the list of references**

e) **[20 points] (Section III): Results for dataset 2**
   a. **Every subsection here will be one of the models**
   b. **Expect clarity on methods (see the first part of this rubric to clarify how to present)**
   c. Every mathematical term introduced should be defined precisely
   d. All figures should be captioned, referenced in the text, have clear axes, legends etc.
   e. **Code**: All code for each dataset should be in the Appendix – separate section for each dataset in the appendix.
      i. Use subsections for each algorithm for each dataset to corresponding code
      ii. The section#.subsection# in the Appendix should be cross-referenced in the main document in the appropriate dataset+algorithm section for ease of search
   f. **ALL CODE WILL BE TESTED.** **<u>Details on how to run it should be in the Appendix</u>. (code taken from author's site or GitHub should only be used to test your code).**
   g. **If code is shared also via a GitHub link, then the code must be made public and a clickable link must be provided in the list of references**

f) **[20 points] (Section IV): Results for dataset 3**
   a. **Every subsection here will be one of the models**
   b. **Expect clarity on methods (see the first part of this rubric to clarify how to present)**
   c. Every mathematical term introduced should be defined precisely
   d. All figures should be captioned, referenced in the text, have clear axes, legends etc.
   e. **Code**: All code for each dataset should be in the Appendix – separate section for each dataset in the appendix.
      i. Use subsections for each algorithm for each dataset to corresponding code
      ii. These section#.subsection# in the Appendix should be cross-referenced in the main document in the appropriate dataset+algorithm section for ease of search
   f. **ALL CODE WILL BE TESTED.** **<u>Details on how to run it should be in the Appendix</u>. (code taken from author's site or GitHub should only be used to test your code).**

g) [**5 points**] (**Section V**): **Conclusions (inc. pros and cons of each dataset and model)**

a. See rubric earlier to clarify what to present here
b. Be precise and succinct

h) **[5 points] Section VI: Value of concepts learned in course**
   a. You must clarify what techniques from the course you used to understand the models and methods
   b. While such a connection may be harder to make with deep learning, you can and should still try to make such connections as best as possible (e.g., learning low dimensional representations, last linear/logistic layer, etc.).

i) [**5 points**] (**Section VII**): **Conclusions**
   a. clarify challenges of each dataset and algorithm
   b. Compare for each dataset the algorithms briefly and briefly identify the best algorithm based

j) **[5 points]** (**Section VIII**): **References**
   a. A list of RELEVANT references should be included. This can also include lecture slides or other papers/books/notes you used to understand the concept.
   b. Points will be deducted if **references are not explicitly referenced in the main text**.
   c. Ideally, references should be listed in the order of appearance; alternatively, they can be listed alphabetically.

k) **PLOTS/Tables (choose wisely based on space, choice etc.) TO BE INCLUDED IN MAIN DOCUMENT**
   a. Some mechanism to highlight hyperparameter tuning or some clarification for how you chose it in different settings
   b. **Essentials:**
      i. Plot of training vs. test loss for select settings to demonstrate your training setup and results
      ii. plot of classification metric vs. model complexity
      iii. All other details that are relevant and can fit in your narrative of about 2 pages per dataset – use a combination of words, tables, plots, sentences with details etc.

l) **Deduction of Points**
   a. 4 points per section will be deducted if pages are not referenced in Gradescope.
   b. Points will be deducted for not enumerating tables and figures and referencing them as Fig. # or Table # and referencing them in the text.
   c. Points will also be deducted for not precisely referencing the page numbers for each section.

m) **Filename convention**: <Lastname1,Lastname2,..._EEE598_FinalProject>.PDF or .zip. Use lastnames of ALL students on project. (Of course, do not use commas to separate last names in actual filename; **use dash**)

n) **Zip files should only have PDF and code files**

o) **Writing:**

   a. Spelling, grammar, and punctuation matter.
   b. Write short easy to read sentences without resorting to using multiple conjunctions and compound sentences with running phrases that are hard to comprehend.
   c. It is OK to use generative AI to help write but honestly about using prompts is essential as this can also be viewed as academic integrity violation.

p) **Code**: If code largely mirrors an existing GitHub repo, the final project will lose 50% of points.

q) **Attachments**: Main report should be uploaded on Gradescope; code on which the report is based should be uploaded to canvas.