CSCE 623 – Project Assignment Notes & Additional writing guidance

General hints when **Running ML Experiments in code**:

* When running an experiment/simulation, it is ok to *collect* your data using something other than python, but once the data has been collected/saved, all further processing should be accomplished using repeatable processes in python.
* Don’t wrangle/edit your dataset directly using an interactive data editor (excel, MATLAB, etc.). Start with a fresh copy of the data and complete all of your transformations and preprocessing using steps using (python) code. This ensures reproducibility
* Ensure you control randomness of stochastic components of your code via managing the random seeds.
* You don’t need to put *all* code from your project in a single python notebook, but you might find it helpful to organize your activities with python notebooks.
  + Just like functions in numpy, pandas, and scikit learn, you can import your own python functions written/saved as .py files elsewhere … in this way you can keep your notebook cells relatively clean (once you know the function is working well elsewhere).
* Organize you code in sections according to a standard ML pipeline. Use lots of comments and markdown info to clearly explain what each component does
* Set up your code so it saves your partial outputs (data and figures) in meaningful, date-stamped filenames so that you can extract the info later without needing to re-run computationally-expensive segments of your code.
* When you are satisfied that your pipeline is complete, restart the kernel (to clear all variables) and run the whole pipeline again to ensure you still get the same results. If there is any difference, resolve the issue and keep rerunning your code *until you can restart & run it two times in a row without differences*. Only in this way can you ensure your process is repeatable & reproduceable.

General techniques for success in **technical writing** in this assignment

* Storyboard your design. Before you start writing, use an outline or index cards, or possibly a digital form such as a whiteboard or PowerPoint (e.g., one slide per section) to architect your drafts. Try to figure out what your key messages and goals are in each section (what you want the message to the reader to be) and state them plainly in your storyboard, even if you don’t end up putting that information in your actual draft.
* Write as you go. As you are running your ML research, capture content such as figures and tables and include them in your document (including descriptions). If you wait until the end to include these things, you will spend a huge amount of time trying to find or reproduce them.
* Use saved figures instead of screen captures. Most python graphics functions (matplotlib, seaborn) have a way to save your graphical outputs to a file. Importing image files are always results in cleaner images in your document than trying to use screen captures of your code outputs. Bonus – if you need to change a figure, just change the code and rerun and your imported file will be updated automatically!
* Revise. Write in several revision steps instead of one monolithic linear effort. If you find you don’t have much to say in a section, then move on and work on something else.
* Proofread after moving text. When moving text using drag or cut & paste, it can be easy to create new problems. Redundant explanations and dead/broken sentences can result when you don’t perform a careful review of the surrounding text. It is often easier to handle these problems when you make them than it is to try to discover them later when you are doing your final review and are trying to get it done quickly.
* In early drafts, establish the research, background and method first. Outline the components of the results as well as intro and conclusion. As the content matures, update and polish the intro and conclusion as one of the final steps.
* Write the abstract last. In later drafts, don’t spend too much time trying to write a great abstract until you have completed the rest of your work, introduction or conclusion.

When you think you are done and are **getting ready to finalize your submission**:

* Take a step back and look at how you want to organize things at the paragraph and idea level. Go back to your storyboard and update it. Figure out where each thing would fit best and reassemble your structure before the final draft. Repeat this process of modifying your report and modifying your storyboard until the two converge.
* If you haven’t looked at your introduction, conclusion, or abstract in a while, it is a great time to review it and make sure it covers all the points you want to cover. Remember that your abstract and intro are the first impression you will leave with your reader. Also remember that some readers will make a decision on whether your paper is worth reading by just reviewing the abstract. Many more readers don’t have time to delve into your papers deeply and may just look at the introduction and conclusion, perhaps skimming math formulas and visuals along the way. Make sure all these components look good, and make sure the reader has an extremely good understanding of what you did even if they don’t read any other parts of your paper.
* Even if you are a great writer, you are probably not as good at spelling/grammar/style as you might think – humans make mistakes! Run a spellcheck / grammar check / style check on your work (if you are using MS Word, you can enable continuous proofing)
* Conduct a word count to ensure you are within the bounds for this assignment (see below)
* Give yourself a mental rest cycle before the final review/proofread of the document. Ideally this means a sleep cycle, but if that is not possible, at least do something else between the last editing session and the final proofread before submission. Go for a run. Eat dinner. Play a game. Talk with someone. Do anything other than look at your writing for at least enough time so that you stop thinking about it.
* **Proofread the final version IN THE FILETYPE YOU ARE SUBMITTING (e.g. PDF)**, or in a printed hardcopy - sometimes it is hard to see things on a computer screen so you may want to print your paper before reviewing.
  + Special instructions for LaTeX users: You wouldn’t write python source code and submit it without running and checking the output, would you? Even if your IDE handles real-time syntax checking, you would still run it, to make sure it produces the correct output, right? *You should think about LaTeX as source code* and the PDF as the output: generate the PDF and **carefully proofread the entire PDF** (don’t just proof the LaTeX source). If you go back and make changes to your source files, then repeat the generate-and-proofread process again to make sure you didn’t introduce unintended side effects.

Regarding word count: There are specific word count ranges for this assignment. The lower bound is a hard margin that helps me know that you’ve spent enough time on the assignment. Writing less than the lower bound is an indicator that you have not done enough. The upper bound is a “soft margin” that helps constrain the maximum amount of time you spend writing and the amount of time it takes me to review your work. The upper bound also helps ensure that you’ve concisely expressed your messages. Exceeding the upper bound because you are not concise can affect your grade.

You should conduct a word count on your final product.

* **Undercount**. If you follow the suggestions below it will be unlikely that you will have too little content (very few students will experience this problem), but if you find that you are still under the minimum word count, a good first thing to do is review your approach/methodology and your results section. You can always add more details in these sections since they tell me what YOU did. You should also look at the task and data descriptions – it is easy to forget to include explanations of your starting point which could make the difference between the reader thinking that this is a trivial problem and recognizing that it is a challenging one.
* **Overcount**. A more likely problem is that you have too much content (you have many more words than the upper limit). If you find yourself in this circumstance, you should address the excess in two stages: macro-level cuts and micro-level stylistic changes.
  + *Macro-level cuts* tackle excess by removing dozens or hundreds of words at a time – whole sentences to whole paragraphs. Try to figure out what is important for your reader and eliminate other things. For example, students often spend too much time explaining things which the reader should already know (e.g. ML descriptions and details of inner workings which are already present in the text). Review your related work and introduction to see if you are discussing things which can be cut.
  + *Micro-level stylistic cleanup* opportunities are more difficult to see because excessively verbose style is ingrained in the way we write. There are several searchable markers of verbose style. You might start by looking at parenthetical expressions and in-other-words phrases or examples (i.e.; e.g.). Searching for commas might also help you recognize you have too many caveats or interrupting phrases in your writing. Look at the length of your sentences too – excessively long sentences might indicate overly verbose writing. Try to revise a long sentence by removing the parentheticals, caveats, explanations, examples and see if it still makes sense. If so, you don’t need the removed text.
  + If you are still struggling with too much content, see the end of these writing notes for some suggestions.

Some examples of things to think about are listed below.

Where to put Definitions and other background info:

* If text descriptions of key ideas are needed to understand the nature of the problem and research, these definitions might go in your introduction or related work section.
* Math definitions/tools/techniques used only in other research (but not yours) go in your related work section ONLY IF NECESSARY to understand some critical component of the other research that you will be discussing later (for example in your results section).
* Put domain-relevant text/math definitions used in your research (even if they are used in other research), in your problem definition or approach/methodology section.
* Don’t spend time discussing background material (like stuff covered in our textbook) in the related work section… instead, put short versions of background definitions of measurements/assessments to the approach/methodology section (with citations). If you are using background techniques not covered in our text, you can go into more depth, but another alternative is to put them in an appendix.

Article Style / Format:

You should use a conference or journal article format for the target publication. If you don’t have a target publication, you may use the generic IEEE style format listed below:

Format: don’t use the AFIT thesis/dissertation format for this report. If you have no particular format needed for a specific conference or journal, use the IEEE format for double-column article style:

<https://www.ieee.org/conferences_events/conferences/publishing/templates.html>

More details on what goes in each section:

ABSTRACT: Think of this as both an attention getter and a high-level overview of your entire effort. This should also be your most polished and revised portion of your document. You definitely want to give the reader enough detail so that they know whether or not your paper is something they are interested in reading. Some people highlight their specific results in the abstract (especially when pushing the state of the art or they got great results), but this is not always necessary.

INTRODUCTION: This section has two goals: 1) to motivate/excite your reader and get them interested in your work; 2) to give them enough information to understand what you did (at an abstract level) and give them a preview of your results so that they could decide whether they want to read about the details. Your introduction must motivate the reader by explaining why the problem is important. Don’t just regurgitate what you said in your abstract – make sure you provide additional depth. A good introduction represents a fairly deep overview of what you accomplished – and if the reader only looks at your intro and conclusion, that should be enough to allow them to engage in a meaningful conversation with you about your work. Include a description of your data, the classes (or range of regression values), the ML / DST pipeline you will using, including model types & tuning methods, and an overview of your performance/results.

RELATED WORK: Think of this section as identifying the research that is “competing” with your research (or prior research which you are extending or improving). The main goal of your related work section is to cover discussion of the other research on this (type of) data and outline areas in the other research that could be addressed/improved with your research. If you have no direct competing research, then consider similar research which is different in one aspect. For example, if you are trying to predict prices for equities in a specific closed-entry market, you might not be able to find any other research in this area, but there is plenty of research on people trying to predict prices in the public stock exchange. In addition to giving the reader enough information about the other research that exists, you should point out the research gaps (flaws/assumptions/limitations) with the other work that you hope to address in your work (but stop short of explaining how you addressed these gaps). Describing these gaps will set up the rest of your paper and make your reader interested in finding out how you filled the gap. Other specific guidance includes:

* Use properly formatted citations. Two common formats are APA and IEEE.
* A bibliography manager such as Mendeley (AFIT’s supported bib manager) will be extremely helpful in managing your references and generating properly-formatted citations and reference sections.
* Citations should be within the sentence containing the content you are citing (not after the end of the sentence)
* Avoid multiple separate adjacent citations – instead of “[1], [2]”, or “[1][2]” either use “[1, 2]” or put text between them like this: “… as presented in their work [1] or the extension of their work by James [2].”
* Avoid citation lumping (single citations bigger than 2 refs) unless absolutely necessary. Lumped citations gives the appearance of bibliography padding. In general, you should give each citation at least a phrase, but important citations can have a sentence, paragraph, or in some cases multiple paragraphs. The exception to citation lumps is when you are introducing a topic and you plan to describe the individual citations and their similarities/differences over the next few sentences (use individual citations associated with the particulars of each effort)
* Do not put background info on machine learning or math techniques in related work. If it is needed to understand what you are doing, you can keep it in the text, either in your approach/methodology section (if short), or in an appendix (if long). If a long explanation is not needed, remove background info, and just state the technique and reference the source (like our book).

APPROACH/METHODOLOGY: Think of this section as a detailed technical description of the starting conditions of your specific project (what is the situation right before you actually did any machine learning work). If someone reads this section, it should provide them enough information to start thinking about how they would attack the problem. This section acts to outline differences between existing related work and your approach/methodology. You should identify challenges with this problem space which you will have to address & tackle (this allows you to set up the approach/methodology section).

This section should also include at least a paragraph about your data. You should clearly identify how many observations you have, how many features you have, and the identity & explanation of each of the features, including whether the features are numerical, categorical, Boolean, etc. If you are doing classification, indicate the percentage of the data (or observation count) for each class. If you are doing regression, you should express the distribution/density of the target values (e.g. via a histogram). You will also need to explain your data wrangling/processing pipeline – how did you get the data from its original form into one that could be sent to a model fitting process. Include tables as needed to organize your explanations. The specific method for describing your data will depend on how you obtained it:

* If you obtained a well-formed data set (the data is already in a matrix with observations in rows and features in columns), describe the source of the data as well as the details of the features and observations (counts, encoding types). Make sure you describe what your labels are if you are doing supervised learning.
* If you are creating or collecting the data from a source you control (for example, from a simulation you are going to run or data generator), describe how it was created/collected and the format it is stored in.
* If you are starting with unstructured or messy data (examples: human-subject performance data; unformatted/raw text; webpage/twitter/Facebook scrape; non-standard data table format; time-series electrical signal; gene-sequence data) then give as much detail about its organization and structure prior to any preprocessing you will need to do.

This section should include at least a paragraph about the machine learning task. Clearly state what kind of problem this is: supervised? Unsupervised? classification? For supervised learning, further state whether the goal is inference or prediction, and whether you are doing classification or regression problem. Describe how many classes there are, or the range and nature of the numerical prediction value in regression problems. If the problem is actually a regression problem being recast as a classification (i.e. using thresholds & bins to define class boundaries) then explain what the boundaries are and your justification for binning the values. If you are dissimilar class sizes in classification problem or non-uniform distribution of regression values, state your method for mitigating this issue in your ML process. Next state the definition of success of your work (e.g. at least 90% accuracy; MSE not greater than 7; select the 4 most influential features; find the 3 best clusters). Definitions of success should either be pulled from the problem domain, or from what is being outlined as good ways of judging success in the related work. If none of these exist, you may determining up your own definitions of success – but try to consider the usage of your machine learning models in your domain when you develop your definition of success. When you define success, be clear whether it is being driven by the domain requirement, related work, or it is an informed decision you made.

Your Methodology/Approach section should also include a description of your pipeline (consider the modified KDD process) and provide specifics explaining how you did data collection, exploration, preprocessing, separation (test v. non-test), model selection & tuning, and performance assessment.

RESULTS: This section presents evidence on the efficacy of your work. If your goal is to get better performance with your ML system than other existing solutions, you will need to show the comparison using statistically-sound processes, and either convince the reader that you achieved that or explain why you fell short. All graphs and tables should be cross-referenced and explained in your text – but don’t just regurgitate what you showed – explain its importance and implications.

CONCLUSION AND FUTURE WORK: Remember to cover your key findings/performance results, but make sure to put them in the context of the bigger picture of what your research can do – to include assumptions and limitations that would have to be addressed. Future work should not be trivial – don’t say things like we tried these 3 algorithms and these other 2 should also be tried. Also, don’t say someone should try collecting more data so you can have better performance. Try to give meaningful steps that could be accomplished by someone really interested in standing on the shoulders of your work. Talk about how you would address the limitations / assumptions / shortcomings with your research – especially if you had some unexpected outcomes that you can think of a solution for.

REFERENCES: Use properly formatted bibliographic entries – depending on what venue to plan to submit. Two common formats are APA and IEEE – if you don’t yet have a venue, use APA for human-subject research or IEEE for systems/computer research. A bibliography manager such as Mendeley (AFIT’s supported bib manager) will be extremely helpful here.

**General technical writing tips:**

Avoid the use of “I” in technical writing. Using “I” in technical writing often shifts the focus from the activity to the fact that you were involved. Also using “I” will tend to cause you to fall into a pattern of describing a set/sequence of things you did. Unlike a lab notebook, or log/journal, a technical paper generally should not be a complete chronological description of everything you tried in your research. Listing everything that was accomplished, in the order it was accomplished makes it harder for the reader to determine the overall direction of the research and the relative importance of the steps. Using “we” reduces these issues, but overuse of we can sound monotonous, and still implies that who did the action is more important than the action itself, so you should also minimize the use of we. You might think “how am I going to maintain the active voice if I am the only one doing this project and I cannot talk about myself??” If you are concerned about overuse of the passive voice, consider this: Technical writing is not a drama or an autobiography – it is technical writing. Take the YOU out of the writing. If you are trying to avoid the passive voice there is absolutely no need to associate things with any person.

For example, don’t say this:

“The data set I have been given contains 3,600 features with only 5000 observations.”

Because nobody cares that YOU were GIVEN the dataset and you are irrelevant in the description. Instead, you can keep the sentence active like this:

“The dataset contains 5000 observations, each with 3,600 features.”

The exception to the use of identifying yourself in the writing would be when referring to previous published/submitted work that you accomplished. In that case, you should have a (self)-citation and reference, and talk about yourself just as any other author: “In another approach, Hefron & Borghetti used random forests to classify the time segments into high or low workload conditions [1]”

Avoid idioms, colloquialisms and conversational simplifications of technical descriptions. For example, instead of saying “broke down” or “crashed”, say “was unsuccessful” OR “did not succeed”. Instead of saying “the output was jumping around too wildly” say “the output was marked by excessive variability”

Help! I still have too much content / too many words – how do I decide what to cut?

* Cut background info: If a common technique is well-described in our textbook (or another source) consider removing it and replacing it with a reference and citation.
* Figure out what your commonly used *empty words* and *unwieldy phrases* are. Hunt them down with the search tool and then cut or re-design these sentences– examples:
  + “As such, the null hypothesis…” could be “The null hypothesis…”
  + “With regards to the bagging method the number of estimators (weak learners) was varied…” could have been written “The number of bagging estimators (weak learners) was varied…”
  + “In order to…” becomes “To…”
  + “The results of the processes <A> and <B> were <X> and <Y> respectfully.” becomes “Process <A> yielded <X> and process <B> yielded <Y>.”