

Final Project Report for CS 175, Winter 2021

Project Name: `project title/team-name goes here`

Student Name(s)

Name1, StudentID1, uci_email_address

Name2, StudentID2, uci_email_address

Name3, StudentID3, uci_email_address

General Instructions:

- **What to submit**
 1. *PDF report (modified version of this document), one per team. Replace "PROJECTNAME" with your team's project name in the title of the PDF file.*
 2. *Zip file with code (see below), one per team*
 3. *Individual contribution report (see below), one per individual*
- *Your report should be 5 to 8 pages long in PDF (not including the Appendices). Remove all blue text and replace with your own text. You can re-use any material (figures, tables, text) from your proposals and progress report if you wish.*
- *If you want to add more details (e.g., additional graphs, examples of your system's output, etc) beyond the page limit, feel free to add such information to Appendices at the end of your report.*
- *The sections where I expect to see most new material (at least for most teams), and where I will focus on the most for grading, are Sections 4 and 5(Experiments and Insights), since these are the sections where you are likely to have the most new material to report relative to earlier reports. Don't ignore the other sections, but pay particular attention to Sections 4 and 5*
- **What the Team submits to Canvas (one student submits on behalf of the team)**
 1. *Report: Your report in PDF format*
 2. *Software: A zip file called PROJECTNAME.zip, where PROJECTNAME is the name of your team's project, that contains the following*
 - *A Python script or a Jupyter notebook (e.g., called project.py or project.ipynb) that can be run directly and that demonstrates at least some important aspect (not necessarily all) of your project. Your notebook can import a sample of the data that you used, import 1 or more models that you built, and generate examples of the types of predictions or simulations your model can make. The notebook should not take any longer than about 1 minute to run in total (if you have models that require a lot of training time, train them offline and just upload the models and some sample data to illustrate them). Feel free to generate examples of your model(s) in action, e.g., for reviews you could*

generate examples of reviews where the models work well and reviews where the models work poorly. Note that if your model or data is too large to upload, you can include a link to the model and/or data in your script or Notebook to an external location that we can download it from.

- *Also save a .html version of your notebook called project.html, showing the outputs of all the cells in the notebook (in case we can't run your notebook)*
 - *Also include a subdirectory called src with all of the individual code (scripts, modules) for Python (or equivalent for other languages) that your team wrote or adapted– these don't need to be called by the project.ipynb notebook but should be made available in the src/ directory.*
 - *A README file that contains a 1 line description of each file in src/*
 - *Note that we don't necessarily plan to run the code in src/, but may want to look and run parts of it.*
 - *Also note: don't worry too much about writing a fancy script or notebook, or spending a lot of time cleaning up your code. Do some tidying up and documentation, but you should spend most of your time on the report.*
- *Individual Contribution (each student submits a separate individual document to Canvas)
Each team member needs to submit a ½ page of additional text (in PDF format) as an individual, with the title "IndividualContribution.pdf" that provides an honest assessment of which parts of the project you contributed to and which parts were worked on jointly. This should be written individually – you may wish to discuss the plan of what you will write with your project partners, but the page you write should be generated separately by each individual. Put your name and student ID at the top of the page. There is no template for this individual document.*

1. Project Summary (1 or 2 paragraphs)

*[This can be the same or similar to what you wrote in your proposal or progress report]
This should be short (2 to 3 sentence) high-level summary of what your project is about.*

2. Data Sets [at least 1 page]

[This should have considerable detail – make sure you include a good description of your data set(s) – figures and tables are strongly encouraged. Can be an updated version of what you wrote before.]

Describe what data set(s) you have used in your project – include references (e.g., URLs) for where you obtained the data.

3. Technical Approach [1 to 2 pages]

[This can be an updated version of what you have written in earlier submissions....]

Provide a description of the techniques and algorithms that made up the core of your project. For example, if your project involved comparing different classification algorithms for document classification then in this section you would list and briefly describe the classification algorithms you used (e.g., naïve Bayes, logistic regression, support-vector machines, recurrent networks, etc). Be as clear as you can about what versions of algorithms you used.

You can include descriptions of preprocessing software, API/crawling code, data cleaning scripts, etc. Also feel free to include descriptions of algorithms and software you spent time on developing but that did not end up being part of your final project for one reason or another.

It may be helpful to show a block diagram that shows how the different pieces of your system work together, e.g., a pipeline of document preprocessing steps, etc.

- *Here you provide a detailed description of your technical approach, about 1 to 1.5 pages.*
- *If you are using multiple different technical approaches you could break this section into subsections. Pay attention in particular to any details that you were asked to clarify in your original progress report.*
- *Feel free to provide a figure that illustrates your pipeline or system with blocks for different components and brief descriptions of each component (e.g., the names of algorithms or methods you plan to evaluate). Make sure it is clear what your pipeline or system is doing, i.e., what each component will do in terms of taking inputs and producing outputs.*

4. Experiments and Evaluation [at least 2 pages]

[This is a critical part of your final report and the section where I expect new material for most projects compared to your progress report (unless you had already completed most experiments at the time of your progress report)]

Describe in detail both (a) how you set up your experiments, including what metrics and methods you used for evaluation (test sets, cross-validation, user studies, manual evaluation (e.g., for clustering) etc), and (b) what results you obtained (ideally in the form of tables, graphs, etc), e.g., comparing the accuracies different methods and baselines. You can also report additional comparative results on sensitivity of your approach to different algorithmic choices, e.g., how did performance depend on vocabulary size? On document length? On whether you removed stop-words or not? Did including parts of speech help? And so on.

Feel free to include qualitative evaluation, e.g., your own observations about the nature of errors your system is making, or what aspects your system is doing well on: and show specific examples if you can (if not everything fits here feel free to put more examples in the Appendix). If it makes more sense to put qualitative/manual evaluation of your results in the Insights section (next section), feel free to do that.

5. Lessons Learned and Insights [at least ½ a page]

[This can overlap with your project report but for many projects you will want to update this you're your overall insights after completing the project]

Discuss what insights you gained from the project. What did you learn about the algorithms you worked with? e.g., what did you learn about their strengths and weaknesses? what results agreed with your expectations? What did not agree with your expectations, i.e., was surprising? What do you see as the specific challenges in terms of the problem you are trying to address (e.g., lack of labeled data, limitations of current algorithms, etc)? If you were in charge of a research lab, what ideas and directions might you invest in over the next year or two to try to make major progress on this problem? Feel free to be speculative in discussing possible future directions.

APPENDICES

Appendix A: Software [at least ½ a page]

Provide a list of the major pieces of project software and their functionality (general input/output characteristics), both for (a) code you wrote, and (b) code from other people that you used. Feel free to put this information in a table if it helps to organize the information this way.

Appendix B: Additional Results (OPTIONAL)

This is where you can provide additional graphs, tables, examples of results that you would like to include in your report but that did not fit in the main section. Points will not be deducted if you do not include this Appendix.