INFO 4390/5390 / CS 5382
Project Handbook

Professors Koenecke and Pierson, Spring 2024

Goal

This project is designed to give you experience with algorithmic fairness through the lens of research. Because this is an open-ended project, we've included several milestones to keep your group on track.

- **Phase 1**: a literature review on the algorithmic fairness topic you'd like to study. What have researchers or journalists uncovered previously? What datasets and/or methods did they use? As part of this literature review, you should also identify and download data sources that you would like to use for Phase 2.
- **Phase 2**: a coding project. You should use your Phase 1 literature review to (a) identify interesting questions about algorithmic fairness, and (b) write code to answer these questions on the data source(s) identified in Phase 1. You are expected to implement basic machine learning algorithms, examine fairness metrics, and report interpretations of your results.
- **Phase 3**: combining Phases 1 and 2 into a full report, including ample interpretations of your code-based analyses and reporting how your results connect to the broader literature.
- **Phase 4**: a powerpoint presentation. You will convey your Phase 3 report via in-class presentations to your peers.
- **Phase 5**: the "final" graded version of your Phase 3 which ideally incorporates feedback received from peers.
  - ***Final survey***: you may also optionally submit an individual survey to provide feedback on your teammates' teamwork over the course of the semester.

Here are some examples of projects which combine the qualitative (lit review) and quantitative (coding) factors we're looking for: bias in urban spatial analysis, criminal justice, healthcare, and a similar healthcare problem. Options for good general classes of projects include: audits of APIs for bias; fitting a model on an existing dataset and analyzing your own model for bias; or analyzing an existing algorithm for various biases.

Project Groups

You will work in teams of 3-5 students; your team may only consist of students from the same campus (i.e., Ithaca-based students should work only with Ithaca-based students; Cornell Tech-based students should only work with Cornell-Tech based students). If you are looking for project teammates, please post on Ed Discussion with a description of the types of fairness problems you are hoping to study.

To help your group work together productively and ensure that everyone contributes, we suggest that your entire team discusses and agrees on a set of policies regarding:

- Decision making. For example: consensus, majority vote, or team captain.
- Communication. Methods of communication, expectations for response times.
- Meeting times. When and where you will meet, procedures for missing attendance.
- Balance of responsibilities. Procedures for ensuring that everyone contributes.
- Conflict Resolution. What you will do when you identify violations of this contract or other problems, and how you will resolve them.
- Availability. When each member will not be available for any reason (coursework, travel, athletics, social events, etc), and how you will work around these absences

Disputes within groups should be handled by the students themselves (not by appealing to professors or TAs).

To collaborate with your teammates, we recommend you use version control software to organize your project work, such as Github. You may use Google Colab notebooks, but be aware that simultaneous edits are often buggy and will yield incorrect merges, resulting in an un-runnable notebook, so be careful to avoid this issue!

Due dates

All project work will be submitted via Gradescope. Please make sure your entire project team is tagged in your Gradescope submission; otherwise your grades will not be recorded.

Slip day policies: You have 5 slip days over the course of the semester which you can use at any time during the term without penalty (for both assignments and projects). You can use up to 3 late days on a single assignment. The only exception is that the final project write-up (Phase 5) cannot be submitted late because we need to grade it in a short amount of time. When submitting project work as a team, **each one of you must use a late day**. Once you run out of late days, you will incur a 20 percentage point penalty cumulated over each extra late day you use.

Phase	Due Date	Submission type	% of final grade
Phase 1 (literature review)	Feb 20, 2024	PDF	10%
Phase 2 (code)	March 14, 2024	IPYNB + zip file of data	10%
Phase 3 (combined literature review & code – draft)	April 16, 2024	IPYNB + zip file of data	5%
Phase 4 (presentation)	April 25, 2024	PDF (of slides)	10%

Phase 5 (combined literature review & code – final version)	May 7, 2024	IPYNB + zip file of data	15%

How to do well on this project

- **Balance execution and ambition.** Grading for this project will be a mixture of the quality of the work and the degree of difficulty. If your project is relatively straightforward to execute, you will need to be more thorough and polished to get a high score; if you are combining three different APIs in complicated formats with slightly mismatching IDs, we will tolerate more messiness.
- **Start early.** Many of your initial ideas may not work. The phases are designed to encourage you to work in stages, but the project will take time and you will get stuck. Leave space to think about problems and find solutions. TAs will have more time to give feedback before the deadline crunch.
- **Work as a group.** Communication and clear expectations sound obvious, but it can be difficult to put these into practice when everyone is busy. The biggest correlation we have seen with group success is the ability to schedule and attend meetings. Do not divide the project into discrete tasks and staple them together at the very end; this (a) never works and (b) is really obvious. Everyone in a group should be contributing to every part of the project to some degree.
- **Don't forget to analyze and reflect**. The strongest projects often differentiate themselves through their interpretation. Less advanced projects often stop with presentation of quantitative results and don't tell us what these results mean, why they matter, or what the limitations are.

Phase 1: literature review

For your literature review and dataset selection, **we want you to pick a topic you find interesting**, and where there may be some quantitative data to analyze. You can try looking at recent investigative news articles covering instances where algorithms have gone wrong (try, e.g., ProPublica, Wired, MIT Technology Review, The New York Times, and examples discussed in class and by the guest lecturers.). You can also pick a specific technology of interest, and use Google Scholar to look for relevant papers and data.

You will want to find (good) data (which will depend on the research questions you end up formulating). This may take some time, and you may not find exactly the data you want in one file or in one sitting. Your interest may also shift, the more you search and realize what kind of data is available within the topic. Be willing to keep looking for (additional) data and iterating on your topic! One key thing to consider is whether your dataset includes a column for a “sensitive attribute” on which you would like to evaluate fairness.

Some examples of datasets in common algorithmic fairness applications include:

1. Criminal Justice: COMPAS, OpenPolicing

2. Lending: German Credit, Home Credit
3. Education: Academic Performance, Admissions
4. Health: too many to list; refer to https://www.altexsoft.com/blog/medical-datasets/
5. Text: Project Gutenberg, Cornell's Convokit
6. Housing: Boston Housing
7. Images: ImageNet

You can also browse general data repositories such as:

- Kaggle (if the dataset poster is an accredited user, e.g. a company or academic institution running a competition)
- https://archive.ics.uci.edu/datasets (similar to our note on Kaggle, make sure the dataset lists credible sources)
- https://data.fivethirtyeight.com/
- https://data.census.gov/
- https://opendata.cityofnewyork.us/
- https://wonder.cdc.gov/Welcome.html

As general guidelines for choosing dataset(s):

- It should be large enough to have interesting complexity, but not so big as to be unwieldy. As a rough guideline, your dataset should be longer than you could print on a single page in standard spreadsheet format, but small enough that you can run experiments on it efficiently.
- You may combine existing datasets, combine data from APIs, or create entirely new data through instruments or surveys.

We expect you to cite the previous work in this space using the ACM citation style. Keeping automatically-generated citations consistent when working in shared group documents can often be tricky; one option is to use citation management software like Zotero to keep track of your citations within your Phase 1 (Zotero can be used as a plug-in to Google Docs).

Phase 2: code

Here, we'll expect you to do some basic data cleaning and analysis. Start by exploring your data with the most basic types of analyses (summary statistics, histograms, scatterplots) to get a sense of the data. What *could* the data tell you? What kind of questions would it fail to answer? At this stage, you may also gather additional / slightly different data as necessary. We expect to see some basic descriptive plots of your data.

Next, write down a few concrete research questions and hypotheses. Where have you noticed potential biases? (e.g., Did you find anything fishy when exploring the data? What do you know about the processes that generated the data?) These research questions can be based on prior research; however, if you want credit for re-implementing other people's analyses, you must cite them and rewrite the code from scratch (do not simply copy their code – this is plagiarism!).

Next, determine which analyses would be most appropriate for the type of data you've gathered. We expect you to build models, analyze them, and test your hypotheses. Don't throw out analyses that fail to show significance; these can still teach you something about the context from which your data came, or the data itself, if interpreted properly.

Finally, you should interpret your results. Make sure to include markdown cells accompanying your code cells in your ipynb that explain your analyses, interpret your results for a lay audience, and comment on whether the results are expected in the context of the broader literature review you previously conducted.

Phase 3: complete project draft

Here, we expect you to combine your Phase 1 and 2 into a fuller deliverable that reads as a focused final report. You can think of this output as something that, if presented to potential employers who are not experts in algorithmic fairness, would both showcase your ability to write a cohesive narrative and execute correct code, and also be understandable to a lay audience. You should include an introduction and literature review, additional information about your data (including a data card following *Datasheets for Datasets*), code interspersed with explanations of your analyses and interpretations of your results, plots and tables as appropriate, data & analysis limitations, and conclusions.

A few notes on writing for a lay audience:

- Help non-experts understand why your results matter: don't assume they'll know this automatically.. What do these results mean for the "real world" beyond your dataset?
- Avoid chart barf: Pick the clearest and most interesting analyses and contextualize them in your final report. Don't just present every potential combination of numbers and plots: explain to the reader what they mean. Answer the question "so what?". Put additional analyses tangential to the final direction of your project in (optional) appendices.
- It's okay to have null results (e.g., you expected two things to correlate, but they don't show statistically significant correlation): do not use p-hacking! As long as your analysis is one that makes sense to run, you will not be graded differently if your results are significant versus non-significant.
- It's okay to have limitations, as long as you explain them: Were you limited in your conclusions because of issues with the data? What were the issues and how did they specifically impact your analyses?

Phase 4: presentation

Each project team will be expected to give a 8 minute presentation on their project, followed by 2 minutes for Q&A about their work. Teams will be graded on the clarity of their slides, ability to explain each relevant section (reflected in Phase 3), and ability to answer audience questions. Because presentations will occur across multiple lectures due to the size of the class, we will expect all presentation slides (saved as PDFs) to be due on the same day (i.e., the day that final presentations begin).

Phase 5: final project

After receiving peer feedback on your Phase 3, we are allowing you additional time to make finishing touches on your project before final submission. This final report will be the one worth the most of your grade.

We will expect to see a Jupyter notebook with executed cells, containing the following sections:

- **Introduction**. What is the context of the work? What research question are you trying to answer? What are your main findings? Include a brief summary of your results.
- **Literature review.** Who has done what research in this space before? What did they find? How does your ipynb relate to this previous work?
- **Datasheets description of data.** This should be inspired by the format presented in Gebru et al, 2018. Answer any relevant questions from sections 3.1-3.5 of the Gebru et al article, especially the following questions:
  - What are the observations (rows) and the attributes (columns)?
  - Why was this dataset created?
  - Who funded the creation of the dataset?
  - What processes might have influenced what data was observed and recorded and what was not?
  - What preprocessing was done, and how did the data come to be in the form that you are using?
  - If people are involved, were they aware of the data collection and if so, what purpose did they expect the data to be used for?
  - Where can your raw source data be found, if applicable? Provide a link to the raw data (hosted on Github, in a Cornell Google Drive or Cornell Box).
- **Data analysis.** Using a combination of markdown cells (to describe what you're doing, or interpret output from code) and code cells (to execute code):
  - Report basic summary statistics and include plots, as useful, to describe your dataset
  - Run models / algorithms that address your research question
  - Evaluate the significance and/or interpret your results
- **Conclusions.** What did you find over the course of your data analysis, and how confident are you in these conclusions? Detail your results more so than in the introduction, now that the reader is familiar with your methods and analysis. Interpret these results in the wider context of the real-life application from where your data hails. Compare your results to those from your literature review.
- **Limitations.** What are the limitations of your study? What are the biases in your data or assumptions of your analyses that specifically affect the conclusions you're able to draw?
- **Bibliography**. Please make sure everything is cited in ACM style.
- **Source code**. Provide a link to your Github repository (or other file hosting site) that has all of your project code (if applicable). For example, you might include web scraping code or data filtering and aggregation code.

- **Acknowledgments**. Recognize any people or online resources that you found helpful. These can be tutorials, software packages, Stack Overflow questions, peers, and data sources. Showing gratitude is a great way to feel happier! But it also has the nice side-effect of reassuring us that you're not passing off someone else's work as your own. Crossover with other courses is permitted and encouraged, but it must be clearly stated, and it must be obvious what parts were and were not done for this class. Copying without attribution robs you of the chance to learn, and wastes our time investigating.
- **Appendix: Data cleaning.** Relegate any data cleaning that was done for Phase 2 to an appendix (make sure to save your cleaned dataset as a .csv that you can import at the top of your final ipynb, on which you can directly run your analyses). Your data cleaning appendix should be a separate Jupyter notebook with executed cells, and it should output the dataset you submit as part of your project (e.g. written as a .csv file).
- (Optional) **Other appendices.** You will almost certainly feel that you have done a lot of work that didn't end up in the final report. We want you to edit and focus, but we also want to make sure that there's a place for work that didn't work out or that didn't fit in the final presentation. You may include any analyses you tried but were tangential to the final direction of your main report. Graders may briefly look at these appendices, but they also may not. You want to make your final report interesting enough that the graders don't feel the need to look at other things you tried. "Interesting" doesn't necessarily mean that the results in your final report were all statistically significant; it could be that your results were not significant but you were able to interpret them in an interesting and informed way.

Tips for producing an advanced final project

We expect the total length to be 1500-3000 words. Inside this range, length will not be a factor in grading.

Introduction: The introduction should be the exposition of the article where you can use less rigorous language. Your language should be generally accessible, clear, concise, and free of spelling and grammar errors. Aim for this to be readable by someone who hasn't taken this class (maybe your roommate, your family, or you at the start of the semester). It should still be formal, but someone should come to the end and want to read more. 538 articles might be a good baseline tone for this.

Datasheet: As described above, in the style of Gebru et al. Think of this as the "origin story" of your data set. Answer all of the questions listed in the previous section. You can write this in any style as long as it's easy to read as a Q&A. Datasheet will be graded on content, not style. Follow sections 3.1- 3.5 (Motivation to Uses) in this article.

Data analysis and evaluation of significance: Here you will clearly detail your methods used in each part. Qualitative claims made in the exposition should have numerical backing here (instead of "X is larger than Y" write "X is 3.65 times larger than Y"). This should read like a scientific paper, but does not need to be "stuffy" or overly indirect: "we did ..." is more natural than "... was done". A reader should be able to replicate your experiments and findings via their own code after reading this.

It’s important to organize your analysis. Common organizational patterns:

- Big to small. Start with a high-level description of the complete dataset, then add more detail and increase specificity until you are looking at individual data points.
- Small to big. The opposite: start with individual data points, then “zoom out” progressively until you get to a broad, top-level overview.
- Bites at the apple. Visit different facets of the dataset. This could be subsets of the observations along different criteria, or a series of aggregate views where you are grouping by different variables (eg alumni by state, then by industry, then by major).

In most cases you will try many possible analyses. While you should be clear about what analyses you did, we do not need to see full details for every single analysis you tried (see above point on “chart barf”). In most datasets there are potentially thousands of different functions that you could analyze. Why are the ones you chose the most interesting?

Advanced analyses will be clear, logical, and methodical. Mathematical modeling will have clear purpose that answers relevant questions and contributes to an overall perspective. Results will be contextualized with significance tests or comparisons to alternative simpler explanations. Reasonable "next questions" should be followed or acknowledged, though you don't have to follow every lead. Beginning analyses will be disorganized and haphazard. They will apply models without context or purpose. They report results without considering whether those results are meaningful or random noise.

Interpretation and conclusions: This section should reflect on what you accomplished and where you might go from here. These can be hard to write without feeling repetitive. The conclusion is a good place to mention things that you tried that did not work, or data that you could not find but that you would add in a hypothetical further version.

Limitations: "Data" is a selective view of the world that may have been produced in any number of limited, skewed, or biased ways. "Models" are exactly that: miniature representations of real processes that capture some essential relationships but eliminate everything else. Identifying the limitations of your work is a critical part of the data science process.

Code: Submit code as notebooks with the cells already evaluated (i.e., run the notebook prior to turning it in). We won't attempt to debug errors. The most crucial part is to comment your code so that we can quickly understand what it does. This doesn't need to be exhaustive, but you should be keeping your reader updated on what's going on every few lines. Some code may be oriented towards pre-processing and data curation, other code may be oriented towards analysis and presentation of results.

Advanced code will be succinct and well-organized, with comments that indicate expected uses and assumptions for inputs and outputs. Repeated tasks will be broken into functions. Variable names will be informative. Points of failure are anticipated and checked for.

Beginning code will be unclear and disorganized, possibly with large sections of unused code. Variable names will be ambiguous or misleading. Comments will be missing or will simply repeat information that is obvious from context. Variables will be short and uninformative.

Grading

Barring exceptional cases, all members of a group will receive the same grade. Group work occasionally leads to disagreements about the level of effort contributed by individual group members. In some circumstances course staff may use logs of GitHub commits to provide some perspective, and in extremely rare cases we may differentiate grades. If you choose not to use version control software, or use one that does not make user history available, we will not be able to make this type of consideration for your group.

In practice, when the TA staff looks at a large number of projects there is a strong consensus about which projects are more impressive. When TAs separately write down numeric grades and then compare, they are usually all within a few points of each other.

A common question is "do I need to ... to get a good grade?"

It's an open-ended project with additive grading. We only give points for what you do, we never take points off for what you don't do. There are many things that we consider difficult (combining multiple datasets, reformatting data, collecting from web pages), so if you find that any of them make sense, we will recognize that in our consideration of how ambitious you are. None of them are required. Your grade will depend on two factors: how ambitious you are (degree of difficulty) and how well you accomplish your goals (execution). Do not ask us "what did I lose points for?", rather ask "what could I have gained more points for?"

What we want you to do is make an argument based on a data set. If the perfect data set already exists, great! You have more time to work on the details of the modeling and the presentation. In many cases the data set you want doesn't exist in the form you are looking for, and you need to do some work to create it. We want you to have tools to do that if needed. But even if you think you have exactly the data you want, you may find that in investigating it you realize that there are additional questions that require more data collection.

Rubrics

Rubrics for each phase will be used by your project mentor (and for some phases, by peer reviewers) to give you helpful feedback on the current state of your project. Use them to self-assess ahead of submission; figure out which aspects of your work are sufficiently advanced, and which aspects could use more work, then spend time on the latter. **We will be posting these rubrics on Canvas ahead of each phase deadline.**

Remember: our priority in grading every preliminary phase of the project (before the final submission) is to give you good feedback that helps you continue to develop your project. We want to help you produce a final project that you're proud of!

Writing help

If you could use more support in your writing, the Cornell Writing Centers are a free resource available to students looking to improve their writing. They can help you with specific assignments (like the final project phases!) and they offer their services online.

—

Many thanks to David Mimno, Irene Papst, and Carlo Tomasi for contributions in writing this document.