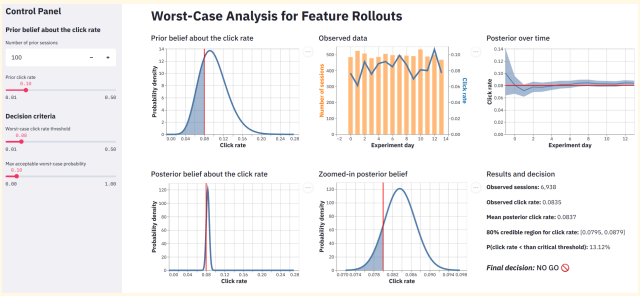
CMSE 830 Projects

This course is project based with two projects: you can think of them as your “midterm” and “final”, since there are no exams.

The projects serve several goals. One of the goals is to ensure that you have something to show at the end of this course. To do this you will write dashboard web applications that are accessible from your personal GitHub account. We will all use Streamlit so that we can share our ideas and skills. In addition to learning to build web apps, share ideas and code through a repository, you will also learn excellent communication skills. You can link to your app from your CV and use your CMSE 830 projects in future job interviews (and your friends and family!).



Nearly all of the work on the projects is done *by you outside of class*. You know already that you will be tempted to procrastinate, so now is the time to think about how to trick yourself so that you don’t do that. I will help you by encouraging you to make steady progress in your homeworks. I highly recommend using a calendar and putting milestones for this course into it with early reminders before due dates.

FAQ

Let’s quickly cover some of the most obvious questions you have about the projects.

- *Do we work alone or in groups?*
  - The default is that all projects are done *individually* because I want to ensure each of you learns everything from start to finish. Of course, feel free to discuss your project with your colleagues and learn from them. But, the projects are yours alone.
- *How much are the projects worth?*
  - The midterm project is worth 10% of your final grade and the final project is worth 20%. Check the syllabus to see how this could impact your final grade.
- *Yeah, but we really want to work together!*
  - If you have a project in mind that is super-sized and it really does take >1 person, then discuss it with me. You will need to write *an additional special proposal* outlining clearly how each person is going to learn an equivalent level of detail, how the work will be divided and what your risk mitigation strategy is. I rarely see this work well and I discourage even trying!
- *What are the projects?*
  - The default is that you design your own project. You are the person becoming a data scientist and you need to think about how to do it independently, another

reason each student does their own project. I will guide you through the homeworks on how to think about and construct a project. Below, I outline some strategies you can choose from. The basic idea is that you use all of the ideas covered in this course: data integration, cleaning, transforming, imputing, visualizing, app deployment, and so on; that is, you will put all of these together.

- *What are the steps?*
    - Keep in mind that there are two major deadlines for the midterm and final projects. As I mentioned above, I will guide you through the thought process via the homeworks; these are the minor deadlines. You will write proposals and have them reviewed, for example. A rough timeline is given below.
    - Because each of you will design your own project, the steps might differ slightly. Obviously, projects that use textual data versus spatiotemporal data don't follow identical steps.
- *Do the two (midterm and final) projects need to be related?*
    - No: you can have two distinct projects. This could happen for several reasons:
        - the midterm project didn't work out as planned and it is time to pivot
        - you had two great ideas and you would like the opportunity to explore both
        - you learned something new and interesting in the class that you didn't know about and that has piqued your interest for the final project
    - Yes: you can make the midterm project extend through the final project to get much more out of your idea. In this scenario, design a large project that you can break into two logical steps; the midterm will illustrate completion of the first logical step.
- *What will I actually turn in?*
    - For both the midterm and final project you will provide:
        - link to your GitHub repository (with all relevant codes and documentation)
        - operational Streamlit app
        - the app should use many Streamlit options: sidebars, tabs, dropdowns, tables, etc.
        - app will demonstrate most of the data science ideas discussed in this course (up to the due date)
        - app must use multiple, advanced interactive visualizations
        - there is no written report, but the app should be completely self-contained in the sense that it is self-documenting and easy for a new user to operate (e.g., there is a documentation page within the app or instructions spread through the pages)

Goal

As we will discuss in the first half of the semester, we will emphasize the ***goal*** of the project more than the technical details. While the technical details are important, knowing ***why*** you are doing something is as important: if you don't know ***why*** you are doing something, you will probably *put garbage in and get garbage out*. You will want to build a narrative first, and then seek the data. You may need to iterate because there are so many datasets available and it is unlikely there is a perfect one for your goal; this is why we start early and don't procrastinate! It's part of the learning process to iterate on several ideas before one works.

This goal will connect directly to the way we will approach visualization. All of your visualizations in this course should begin with what your message is, what story you are trying to tell, what point you are trying to make. Always start first with your narrative before doing any data science or visualization. Do not put random, useless visualizations into your projects/apps.

Projects for PhD Students

If you are a PhD student you are invited to use CMSE 830 as a way to get a great chapter into your dissertation! You probably have interesting data you are already working with and you can use that for your project.

Some projects can lead to publications. This is a realistic goal if you have the data and interest.

There is an alternative approach for PhD students. Perhaps you are taking this course only because you are interested, not because it connects strongly with your thesis research. Or, you would rather have fun with other topics in this course to get your mind off of your thesis. Great - pick another topic of interest to you!

If you want to talk through this, please come to office hours and let's chat!

Projects for Master's Students

If you are in MSU's MSDS program, you likely don't have a project already in mind. If you do - great! If you don't, I recommend thinking of a topic that interests you: you will enjoy the project much more if you are passionate about what you are doing.

There are some common topics people choose, which might help you think about what you are interested in:

- sports
    - there is a lot of data on certain sports, such as baseball and soccer
- finance
    - everyone wants to make millions, right?
- medical

    - dashboards that allow you to understand your health or design a diet
- physics
    - astronomy/astrophysics are popular, but often don't lead to great projects (unless you have a large-enough dataset)

Datasets

Use the internet to find datasets. Perhaps the two best are:

- Kaggle
- UC Irvine Machine Learning

I recommend exploring these two websites to come up with several possible ideas for your project before settling on a specific choice: it would be convenient if your passion overlapped with existing datasets.

The best projects do not use a simple dataset, such as the classics (e.g., iris, penguins, MNIST), but integrate several datasets together. Think about building a unique dataset that combines information from several sources.

Keep in mind that if your dataset is too simple, you will not be able to use the techniques of this course. For example, you can't demonstrate your knowledge of missingess if you start with a dataset that has no missing information. Seek out and build difficult datasets.

Special Projects

If you are struggling to find a good topic for your project, I have some ideas for you.

Teaching Data Science to the Public: Hands-On Data Science

One area I would like to explore is data science education in the context of interactive applications used by the public. The goal is to build apps that could be used at a museum or science fair. The projects would be saved "forever" in Github for anyone to use the code as part of their outreach activities. Future students should be able to access your code for use in real science fairs. Several data science areas will be explored each with an interactive interface that allows visitors to play with the data in interesting and educational ways. Examples might include:

- sentiment analysis: the user enters words and a visualization ranks/clusters what it sees
- recommender system: the user chooses several movies and the system makes recommendations, with allowance for the user to add and subtract movies and see the outcome
- forecasting: teach the public how past data is used to predict, for example, the weather, stock prices, etc.

The app would both show what the underlying algorithm is doing in a way kids can understand and visualize the result.

Apart from the proposals you will write and review, this choice requires a separate proposal early in the semester. Final code open to the public at GitHub.

Adventure Game Using a Large Language Model

The goal is to write and open source a Python/Streamlit-based text-based adventure game. This could be in the style of this ancient game. The goals would be to develop an interactive web game, visualization, and a deep focus on language and text. You will use regex to parse user inputs and the LLM to create the adventure itself. This requires using, for example, the API for GPT-4o. This project is aimed at students who are interested in text as data. As we don't cover textual data until late in the course, early self-learning is required. A fully-operational, publicly accessible app is required by the final project due date.

Apart from the proposals you will write and review, this choice requires a separate proposal early in the semester. Final code open to the public at GitHub.

Flexibility

The rubric will be detailed in the next two sections. Feel free to follow these instructions to the letter.

But, as you have read above, there is some flexibility to allow you to explore your own interests. It is therefore impossible to write a rubric for all possible projects. In cases where the project doesn't match the rubric given below, we can find the equivalent steps for your project. Typically this is done by taking the ***spirit*** of the step and asking how that spirit applies to your project. Once we see your project proposal we can generate an equivalent rubric. See the note below the rubrics for more information.

What is most important is that you understand what your project is, what are your goals, what mistakes were possible, how did you avoid mistakes (mainly in thinking), what you did about errors in the data and why, and can you effectively communicate what you did and why you did it? Anyone can run Python libraries and get reasonable results - boring! But, not everyone can lead a data science project from top to bottom.

Midterm Project Rubric

Base Requirements (80% - B grade)

1. **Data Collection and Preparation (25%)**
    - Use at least two distinct data sources
    - Perform basic data cleaning (handling missing values, removing duplicates)
    - Demonstrate understanding of data types and encoding
2. **Exploratory Data Analysis and Visualization (25%)**
    - Create at least 3 different types of visualizations
    - Provide basic statistical summaries of key variables
    - Demonstrate appropriate use of data encoding in visualizations
3. **Data Processing (15%)**
    - Implement at least one technique for handling missing data
    - Demonstrate basic imputation techniques
4. **Streamlit App Development (25%)**
    - Create a functional Streamlit app with at least 2 interactive elements
    - Include basic documentation within the app
    - Deploy the app and make it accessible online
5. **GitHub Repository (10%)**
    - Maintain a well-organized GitHub repository with clear documentation
    - Include a README file with project overview and setup instructions

Above and Beyond (Additional 20% - A grade)

Students can earn additional points by implementing any of the following:

6. **Advanced Data Techniques (Up to 5%)**
    - Implement advanced data cleaning techniques
    - Use more complex data integration methods
7. **Sophisticated Analysis and Visualization (Up to 5%)**
    - Create advanced, interactive visualizations
    - Perform in-depth analysis of data distributions and relationships
8. **Advanced Data Processing (Up to 5%)**
    - Implement multiple imputation techniques and compare their effectiveness
    - Demonstrate handling of complex missing data patterns
9. **Enhanced App Features (Up to 5%)**
    - Implement more advanced Streamlit features
    - Create a polished, user-friendly interface
10. **Project Complexity and Originality (Up to 5%)**
    - Tackle a particularly challenging or unique problem
    - Demonstrate exceptional creativity in approach or implementation

Final Project Rubric

Base Requirements (80% - B grade)

1. **Data Collection and Preparation (15%)**
    - Use at least three distinct data sources
    - Perform advanced data cleaning and preprocessing
    - Demonstrate complex data integration techniques
2. **Exploratory Data Analysis and Visualization (15%)**
    - Create at least 5 different types of visualizations, including advanced types
    - Provide comprehensive statistical analysis of the dataset
3. **Data Processing and Feature Engineering (15%)**
    - Implement multiple feature engineering techniques
    - Demonstrate advanced data transformation methods
4. **Model Development and Evaluation (20%)**
    - Implement at least two different machine learning models
    - Perform thorough model evaluation and comparison
    - Demonstrate understanding of model selection and validation techniques
5. **Streamlit App Development (25%)**
    - Create a comprehensive Streamlit app with at least 5 interactive elements
    - Include detailed documentation and user guide within the app
    - Implement advanced Streamlit features (e.g., caching, session state)
    - Deploy the app and ensure it's robust and user-friendly
6. **GitHub Repository and Documentation (10%)**
    - Maintain a professional-grade GitHub repository
    - Include comprehensive documentation, including data dictionaries and modeling approach

Above and Beyond (Additional 20% - A grade)

Students can earn additional points by implementing any of the following:

7. **Advanced Modeling Techniques (Up to 5%)**
    - Implement advanced algorithms (e.g., deep learning, ensemble methods)
    - Demonstrate sophisticated hyperparameter tuning and optimization
    - *because this isn't a machine learning course, some of you may be less familiar with these techniques: no problem, please come to office hours for more information*
8. **Specialized Data Science Applications (Up to 5%)**
    - Successfully apply techniques for specialized data types (e.g., text, time series, graph, audio)
    - Demonstrate proficiency in domain-specific methodologies
9. **High-Performance Computing (Up to 5%)**
    - Implement techniques for handling large-scale datasets

    - Demonstrate use of parallel processing or cloud computing resources
10. **Real-world Application and Impact (Up to 5%)**
    - Clearly demonstrate the real-world applicability of the project
    - Provide insightful conclusions and recommendations based on the analysis
11. **Exceptional Presentation and Visualization (Up to 5%)**
    - Create publication-quality visualizations
    - Develop an outstanding presentation or demo of the project

Note:

- The total grade can exceed 100% if a student excels in multiple "Above and Beyond" categories for both projects. Extra points can be used to increase a poor homework grade (as judged by the instructor and TA).
- The “Above and Beyond” should also be used when your project does not meet one of the base requirements; in those cases, you should do more of the “Above and Beyond. For example: your dataset is so rich that there is no need to find and integrate several datasets; since you are skipping that learning goal, you will replace it with, say, a high performance computing element.