

CPSC 222 Intro to Data Science

[Gonzaga University](#)

[Gina Sprint](#)

Quantified Self Project

Learner Objectives

At the conclusion of this project assignment, participants should be able to:

- Identify and collect a dataset related to yourself
- Mine a dataset and write-up the insights gathered from the results
- Write up and present the project results

Prerequisites

Before starting this project, participants should be able to:

- Implement data science concepts covered in CPSC 222 using Python

Acknowledgments

Content used in this assignment is based upon information in the following sources:

- None to report

Overview and Requirements

Please see [DA1](#) for an introduction to the "Quantified Self" project. The overview of the project timeline is as follows:

1. Brainstorm the scope of your project and dataset(s)
2. Demo progress on your project, including cleaning, EDA, and at least one hypothesis test (demo due any time 4/22-4/29)
3. Present your project results (presentation on 5/1 during class time)
4. Turn in your project as a Jupyter Notebook (source code and project report). Github link due 5/2 @ 11:59pm... see Project Submission section below for more information

Project Overview

For the "Quantified Self" project, you are going to create and analyze your own dataset by collecting longitudinal data (data collected over time) on yourself. You may work with one other classmate on this project, so long as their longitudinal data is in the same domain as yours and

is comparable. If you choose to work with a partner, I do expect a more in-depth overall project from the two of you, since there are two of you!

Previous DAs have included incremental tasks related to exploring what data on yourself you want to collect and some preliminary analyses of this data. Now, you are going to decide on the dataset, analyses, and hypotheses.

Choose the Dataset

Your "Quantified Self" dataset can be any dataset that includes your own data, so long as it conforms to the following requirements:

1. It spans at least two months of a recent data collection period. Data needs to be sampled at least daily, which would produce a table with at least 60 rows (AKA instances)
2. It has at least 5 attributes of different measurement scales, including a "class" attribute to be used for classification (e.g. the data is labeled and can be used with supervised machine learning algorithms)
3. It contains at least two tables that can be joined (e.g. each table is from a different data source. For examples of this, consider:
 - a. My YouTube Analytics CSV file and the days of the week CSV file from DA3)
 - b. The API data you explored in DA4 that was possibly related to your project

Include your dataset files in your Github repo.

Mine the Dataset

For this part of the project, you will need to perform data preparation (as needed), exploratory analysis of the dataset (including visualizations), statistical analysis of the dataset, and testing/evaluation of a kNN and a decision tree classifier. Your kNN and decision trees classifiers will be graded on how well they work (or if they perform poorly, how well you can explain why you think this is the case).

Be sure to separate utility code from your project-specific code. Put the project-specific code in the Jupyter Notebook and the utility code in file(s) like `utils.py` (more details on this in the "Technical Report" section. Be sure to provide a [README.md](#) describing your project, how to run your project (including any dependencies), and how your project is organized.

Technical Report (Integrated with Source Code in Jupyter Notebook)

Your report should be a narrative that is written in a data-storytelling format. You must include a project title, your name (and your partner's name if you worked with someone), and the class information (CPSC 222, Spring 2025). Your technical report must be spell-checked and free of grammar errors (proofread your writing!). Finally, it must be organized, clear, concise, and easy to understand and follow, with code cells that mostly call functions defined in `utils.py`. This means your Notebook should not have long code cells, but rather short code cells that call code in `utils.py`. Surrounding a code cell you should have narrative describing the input and the code. After the code cell, you should have narrative describing the output and your insights.

Your Notebook should have, at a minimum, the following sections (in the order given below):

1. Introduction: Briefly describe the project domain, the dataset, your hypotheses, and the classification task you implemented. More specifically:
 - a. Why is the domain important to you and why you are researching in this domain
 - b. What is the dataset format (e.g. CSV files, JSON files, a mix of the two, etc.)
 - c. What tables (emphasis on the plural here) are included in the dataset
 - i. How is the data in each table collected
 - ii. How many instances are there in each table
 - d. Include a brief description of the attributes
 - e. What are you trying to classify in the dataset
 - f. What are potential impacts of the results
 - g. Who are stakeholders interested in your results
2. Data Preparation: Provide details about the dataset and data preparation. More specifically:
 - a. What cleaning of the dataset did you need to perform (e.g.. are there missing values and how did you handle the missing values)
 - b. How are you merging the tables
 - c. What are the challenges you experienced with data preparation
3. Exploratory Data Analysis: Provide details about data visualization and statistical analysis. More specifically:
 - a. What data aggregation techniques are you applying (and why)
 - b. What summary statistics you are using (and why)
 - c. What visualizations informatively present the attributes and relationships
 - d. What statistical hypothesis tests are you computing
 - i. Make sure you set your null and alternative hypotheses up correctly.
Please come see me if you have questions about how to do this
4. Classification Results: Describe the classification approach you developed and its performance. More specifically:
 - a. What attribute are you using as class information (i.e., what attribute or attributes are you predicting)
 - i. What is the distribution of the class labels? (e.g. 50% yes, 50% no; or 70% weekday, 30% weekend, etc.)
 - b. What are your hypotheses about the predictions
 - c. How are you evaluating the performance of your kNN and decision tree classifier? How do their results compare?
 - d. What are the challenges you experienced with classification
5. Conclusion: Provide a brief conclusion of your project, including:
 - a. The classification approach you developed, your classifiers' performance, and any ideas you have on ways to improve performance.
 - b. Describe the potential impacts of your work (including ethical impacts) for the stakeholders you described in the introduction.
 - c. Be sure to include sources for your dataset, tutorials followed, Github repos used as references, AI tools used, etc.
 - i. Note: code you submit for your project should be your original code!!

Present the Project

Each team will present their project as a 5-minute poster presentation to a group of about 3-5 people. Your poster will be a single, high-resolution 16:9 slide you make in presentation software like PowerPoint or Google Slides. You can use this example [Poster Slide Template](#), or make your own. You will “present” your poster by connecting to a TV in Bollier 005 or 004 via your laptop and an HDMI cable. Be sure to practice with one of these TVs and make sure you are between TBD and TBD minutes. After your presentation, there will be ~2 minutes of Q&A from the people listening to your presentation at your poster. The presentation should include key information and insights from each of your Jupyter Notebook Technical Report sections.

In addition to presenting your project, you will peer review your fellow students' projects and presentations. The peer review [Google Form](#) is available in the Google Drive Project folder. See the [Project Presentation Schedule](#) document in the Google Drive Folder for which team projects you are assigned to peer evaluate (everyone will evaluate 4 teams). Everyone will also be evaluated by the instructor and/or another faculty/staff member of the university.

Project Submission

Submit the project by providing a link to your public Github repository in the Project Presentation Schedule document on the Project directory on Google Drive. Your Github repository should contain, at a minimum, your input dataset file(s), your Jupyter Notebook, utility Python files, your output result file(s), and a [README file](#).

Note: if you don't want to make your project's repository public, you can accept this private assignment for Github classroom: <https://classroom.github.com/a/3u5GqSwR>

Grading Guidelines

This project is worth 120 points in the DAs grade category. Your assignment will be evaluated based on a successful execution in Jupyter Lab (using the Anaconda Python Distribution v3.12) and adherence to the program requirements. We will grade according to the following criteria:

- 15 pts for mid-project check-in
 - 5 pts for demoing dataset loading and cleaning
 - 5 pts for demoing visualizations
 - 5 pts for demoing preliminary hypothesis testing results
- 10 pts for dataset adhering to project requirements
 - 3 pts for dataset spanning at least two months of a recent data collection period
 - 3 pts for dataset having at least 5 attributes of different measurement scales, including a "class" attribute
 - 4 pts for dataset containing at least two tables from different data source
- 70 pts for implementation
 - 5 pts for clean Github repo (including dataset files, utility .py files, etc.)
 - 5 pts for including a [README](#)
 - 5 pts for Jupyter Notebook Introduction section adhering to project requirements
 - 10 pts for Jupyter Notebook Data Preparation section
 - 10 pts for Jupyter Notebook EDA section (useful and clear data visualizations)

- 10 pts for Jupyter Notebook EDA section (statistical analyses/hypothesis testing)
 - 10 pts for Jupyter Notebook Classification section
 - 5 pts for Jupyter Notebook Conclusion section
 - 10 pts for adherence to the course [coding standard](#)
- 5 pts for relevance/originality of the quantified self project idea (or in other words, how “into” the project did you get? Have fun with it!! 😊)
- 20 pts for presentation
 - 10 pts faculty evaluation
 - 10 pts average of peer review evaluations