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 loading, cleaning, visualizations, and preliminary hypothesis testing results (demo due any time 11/28-12/5)
3. Present your project results (presentation on 12/13 during the final exam time block)
4. Turn in your final project as a Jupyter Notebook (source code and project report). Github link due 12/13 @ midnight… 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 it works (or if it performs 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, Fall 2022). Your technical report must be spell-checked and free of grammar errors (proof-read 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 Analysis: Provide details about the dataset, data preparation, exploratory data analysis, and statistical analysis. 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 challenges with data preparation
    d. What data aggregation techniques are you applying
    e. What visualizations informatively present the attributes and relationships
    f. 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
3. 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 performance of your kNN and decision tree classifier? How do their results compare?
    d. What are challenges with classification
4. Conclusion: Provide a brief conclusion of your project, including:
    a. A short summary of the dataset you used
    b. The classification approach you developed, your classifiers' performance, and any ideas you have on ways to improve performance.
    c. Describe the potential impacts of your work (including ethical impacts) for the stakeholder's you described in the introduction.

Present the Project

Each individual student will present their project during a 3-minute presentation (timer will cut you off at 3 minutes) plus ~1 minute of Q&A. Each pair of students will present their project during a 5-minute presentation (timer will cut you off at 5 minutes) plus ~2 minutes of Q&A. The presentation involves showing your Jupyter Notebook report and should include notable (yet brief) information regarding each of the above sections of your report. Throughout the presentation, be sure to:

- Describe any key components of the code
- Include sources for your dataset, tutorials followed, Github repos referenced, etc.
    - Note: code you submit for your project should be your original code!!

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.

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/QqiPmH_o

Grading Guidelines

This assignment is worth 100 points. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 15 pts for mid-project check in
- 40 pts for implementation
    - 10 pts for relevance/originality of project
    - 25 pts for technical rigor and complexity
    - 5 pts for including a README
- 25 pts for technical reporting in a Jupyter Notebook, including adherence to the course coding standard
- 20 pts for presentation
    - 10 pts instructor evaluation
    - 10 pts average of peer review evaluations