Erin Scott Adams
HCI 584
Summer 2024

Updated Project Spec – Personality Types Data Visualization

Updates highlighted in yellow.

General Project Description
This app will be a data visualization of personality type data using the HEXACO traits framework (https://hexaco.org/). The primary use case for this app will be informational for the general public to learn about different personality types and see the commonalities/differences among them.

The main application flow is to pull in data from https://github.com/salastro/hexaco-person/blob/main/README.md, sort it within my app, and display common themes in a visual format, in which the user can toggle which trait data they are viewing.

This project will be a web GUI, specifically Python, data visualization library Plotly (https://plotly.com/python/) and its companion web UI called Dash (https://dash.plotly.com/).

User Task Vignettes

1. User lands on homepage
    a. Homepage includes some header text and description blurb of the app
    b. A dropdown input field is also visible
    c. Technical details
        i. /homepage route developed using Dash
        ii. HTML & CSS to style input and text
        iii. Vanilla JavaScript to capture user selection event
2. From the dropdown, user selects an option
    a. Options will be something like these 6 HEXACO Personality dimensions:
        i. Honesty-Humility
        ii. Emotionality
        iii. eXtraversion
        iv. Agreeableness (versus Anger)
        v. Conscientiousness
        vi. Openness to Experience
3. Upon selection, data visualization appears, displaying some sort of comparison (like how often the Honesty-Humility trait appeared in the data in relation the median prevalence in the source data)
    a. Visualization options:
        ~~i. Pie Chart https://plotly.com/python/pie-charts/~~
            1. Learned that Pie Chart isn't the best layout for this data – not comparing one piece to the whole

        ii. Bar Chart https://plotly.com/python/bar-charts/
    b. Technical details
        i. This is where the real meat of the app is
        ii. Might need a loading indicator in the UI while data is processing
        iii. Process the data using Python
            1. Filter data relevant to user's selection
        iv. Output the data into visual format using Plotly
            1. Configurations needed to get Plotly to work, intend to follow steps in Plotly's documentation
        v. Hook up Plotly output into Dash app
            1. Display visualization(s) beneath dropdown on homepage

Technical Details

As an initial starting place, I could develop this without Flask and just focus getting to know the data within a Jupyter notebook. This will involve:

- Pulling the data into my project repo (via cloning/forking https://github.com/salastro/hexaco-person/blob/main/src/model.py),
- Creating core functions to manipulate what's coming in from data source
    - Create a "Trait" OOP class that contains method processUserSelection that cleans the data and preps it for Plotly's required format
- Print out stats for the selected trait via the command line

Additional Technical Details

- Data structures
    - Dictionary that stores attributes for each of the personal trait buckets
        - Drill into / loop over these dicts to pull out specific information as needed based on user's selections
- Config file for Plotly configurations
- Application flow detailed in User Task Vignette's section above

Possible Enhancements

- Depending on how working with Plotly goes, could pursue a more complex visualization and/or multiple visualizations
- Won't be doing this:
- ~~Could build out more of an interactive Q&A UI for real-time user inputs~~
    - ~~Using the questions included here https://github.com/salastro/hexaco-person/blob/main/src/model.py~~
    - ~~And then display personalized data based on user's answers~~
- Could develop multiple pages / routes within the app

Final (Self) Assessment

After working through the spec, what was the biggest or most unexpected change you had to make from your sketch?

My biggest change from my sketch is pinpointing what dataset to use. In my sketch, I linked to this possible option: https://www.kaggle.com/datasets/datasnaek/mbti-type. Upon further inspection, this output is much more language-focused than what will work for the app I have in mind. It's hard to skim datasets at a high level and understand what's really happening!

How confident do you feel that you can implement the spec as it's written right now?

I'm feeling like a 7 out of 10 level of confidence that I can implement this.

What is the biggest potential problem that you NEED to solve (or you'll fail)?

I need to nail down what valuable insights does this dataset provides. What do I want to say with this data? I plugged the info in https://github.com/salastro/hexaco-person/blob/main/src/model.py into ChatGPT and asked it to tell me about it, which was helpful. In the domains_questions dict, each of the 100 questions is mapped to one of the traits. There's also a mean, standard deviation, and color included for each trait, which will be helpful for visualizing and making comparisons.

What parts are you least familiar with and might need my help?

Is it okay that by using data in https://github.com/salastro/hexaco-person/blob/main/src/model.py, it will be "hardcoded" in my project repository (vs. pulling in data from an external API)? I also found this public personality assessment API called Traitify: https://publicapis.io/traitify-api/. But this seems more like a personality test itself vs. the outcome/data set based on their official docs https://app.traitify.com/developer. But if the goal is to hook up to an external API, this could work. Looks like I might have to pay $20 for access, which would be okay if this is preferable as a data source.