Michael Sanchez

CPS 3320: Python Programming

Robert Domanski

2/26/20

Project Assignment #1

**Background**:

Sometimes, We all crave a good sandwich. A classic PB&J, Grilled Cheese, Turkey Cheese, or Tuna sounds appetizing. But what if we want something either slightly different or maybe something random based on our own preference? My project is a Sandwich Recommendation Program which will provide at least three ‘Recommendations’ based on either a variation of a basic/primary sandwich type (Meats, Sugars, Cheese) changing the various secondary/condiments(Veggies, Fruits, Salt/Pepper) while also taking in account different Breads or a nearly completely random one based on a User’s preference.

**Hypothesis**:

If I want recommendations for a similar sandwich to a Turkey Cheddar Lettuce Tomato on Whole Wheat Bread, I would want At least 1 recommendation for something added( mayo is added and maybe pepper), A subtraction (Tomato is removed), and one or more things changed/replace( such as Whole Wheat to a Plain Bagel). The Primary element/type should be the least affected (if it starts as Turkey, it should most likely end as a Turkey). The Program should also be able to make ‘rational’ sandwich-related decisions based on compatibility and Likes/Dislikes.

**Data**:

The Data will consist of Dictionaries (and/or Nested Dictionaries) that will contain data between three categories (Bread, Primary, and Secondary/Extra/Condiments). Within each category there will be data that tracks likes/neutral/dislikes(ranked) and a possible compatibility ranking between items.

The Data will represent the options that the recommendation program can choose from, the item’s the user may want to avoid or may want more often and a variable to decided whether two items are compatible (such as: Peanut Butter & Fruits: High compatible vs PB & Ketchup: Low).

The Data will be curated from online sources, personal experience and User Input. The Source of Data such as the food items will have to come from some online list of Sandwich related items, restaurant/shop menus and personal experience.

Possible Biases include my personal experience (which will most likely set the default for compatibility ranking such as PB & Mustard being Low in the ranking). Other biases could come from internet source when seeing an opinion on whether one bread/meat/etc. is compatible or incompatible with another.

Potential Issues include most likely a small set of data being implemented due to a lack of a definitive compatibility ranking.

**Analysis Setup**:

Data for the dictionaries of items w/ defaults(Neutral towards all items and compatibility based most likely off personal/online opinion) will have to be imported from a txt file created before programming begins.

The program will begin by prompting the user with three options: Load Default settings enter 1, Load Personal Setting enter 2 and Create New Settings. Creating New Settings run the user through a setup that uses data from the default file, asking for a first name, run them through the list of items to gather likes/neutral/dislikes and save it under Personal settings txt file. Directly after setting up a user will brought to the main program asking whether they want a recommendation from a preset, random safe or random wildcard(compatibility and Like/Dislike are ignored). From there the program will be focused on making recommendations through checking compatibilities, giving priority to liked items, and subtracting/adding/replacing items with all of this most likely being done through either recursion or loops.

Results will be placed into a List and ‘Sandwiches’ will be printed out separately vertically.

End Result will hopefully show that the program is capable of ‘making’ sandwiches that add, subtract, and/or replace items while taking in account of the primary ingredient, compatibility and likes/dislikes.

**Potential Roadblocks**:

Potential Roadblocks include:

- lack of data(items) for a comprehensive program (due to high probability of having to develop and create a ‘hand-made’ compatibility list/chart of all items).

-for this project and lack of experience with python GUIs, user input will have to be implemented through the keyboard which could lead to user’s entering incorrect inputs.

-Comparisons may be trickier than intended, avoiding/skipping a comparison of all items; figuring if a comparison can be transitive: if x is compatible with y and y is compatible with z thus x is compatible with z.

-attempts at creating recommendations that use two primaries( treating a primary item as a secondary)