Assignment-4 Visual Analytics 690V

First Name: Sanjay Reddy; Last Name: Satti; Spire ID: 31315112

Github Username: Sanjay-Reddy-S
Github Link: https://github.com/Sanjay-Reddy-S/Visual-Analytics-Assignments/tree/master/HW4

On Python2 all the 3 graphs will come up but on Python3 version, one graph is not being displayed (Refer: Piazza Link. The food trends of patients is commented on Python3 version. One can always directly open the html file). Both the versions of .py along with the rendered interactive .html files are present in the repository.

Besides online sites, I've also referenced Pearson's Introduction to Data Mining textbook (Especially for Market-Basket Analysis)

Note that 54 records is too small a dataset to actually infer anything. And also, 1000+ variables are too many dimensions and almost ½ of them have ambiguous definitions. Wherever assumptions are made, I've written it explicitly

There are 3 main parts

- Market-Basket Analysis
    - Since the initial fields are mainly categorical, I used Market-Basket Analysis to find out rules of generation.
    - Support of an itemset (supp(X)) is defined as the number of times it appears in a transaction
    - Confidence of a rule (X->Y) is defined as supp(X∪Y)/supp(X). It is the probability measure of how often itemset 'Y' occurs given itemset 'X' has occurred.
    - Lift of a rule (X->Y) is the ratio of the observed support to that expected if X and Y were independent. If Lift is 1, the variables are independent and it doesn't matter how good the confidence of the rule is. Higher the lift, better the dependence of 'X' and 'Y'
    - I've coded all 3 measures Support, Confidence and Lift.
    - During preprocessing, all the initial categorical variables have been converted from 'Yes' and 'No' to 1 and 0. For 'belly_button' outie has been chosen as 1. Few fields like 'has_pet' have been added to the dataset by combining both the fields of 'cat' and 'dog'
    - Few results: (Reiterating it again, 54 records is too small. Some of the results are true just because of statistics and might not hold any relevance in the real world)

Rule	Confidence	Lift
Smoke_Often -> Cancer	1.0	1.928

Has_pet -> Quit_Smoking	0.286	1.402
Outie_belly -> Diabetes	0.6	2.159
Jewish -> Dems	1.0	1.317
Atheist -> Dems	0.806	1.061
Cancer -> Quit_Smoking	0.25	1.227

Few more results can be seen by running the python file. Any custom rules can also be easily checked by using the functions I wrote.

- Oil and Food trends for patients:
    - For oil trends, I've plotted the various types of oils used with regards to what disease the person has (IF any). As one can easily see from the plot, Olive Oil is universally used by patients (hence the spike in red). Surprisingly, not many people who are healthy use it (hence the dip in green). A similar trend is observed with respect to Fat_Butter.
    - Since oils are also binary data, I've written my own functions (correlate and return_col) to find the patterns. If there any Custom oil trends one can test using the same functions
    - For food trends, I've assumed, all the 'FREQ' variables stand for frequency that person eats that food item per week and 'QUAN' variables stand for quantity that person eats in one sitting. Since, I didn't want to deal with 2 variables, I've multiplied both and stored the result in the 'QUAN' variable for ALL the food items.
    - I've calculated the mean of the various food quantities for both patients and healthy people. For ease of visualization, I've selected only few food_items (I've picked the ones having high numerical value. Another viable strategy would be picking those variables which have considerable difference between both the categories: 'suffering from disease' and 'not'). These fields can be easily customizable by modifying just few lines in the program
    - Few results: As one expects, people who suffer from cancer eat more broccoli than those who don't. People who don't suffer from diabetes eat more peanut butter than those who do suffer. There's considerable difference between the amount of green salad consumed by people suffering from heart ailments and those who don't.
- Nutrition Summary:
    - Since there are many variables like TOTAL 'Vitamin D', 'Vitamin E', 'Water' etc per person, I wanted to calculate how much of these nutrients are present per food item.
    - Since the variables VASTLY outnumber the records, I've chosen only 54 food_items and tried to calculate PER food_item how many nutrients are there.

- I’ve used numpy *linalg* function, to solve this set of linear equations (54 unknowns and 54 equations, result in a unique solution). Note that the values might not be correct, because we don’t accurately know whether this Vitamin quantity is per meal or or per day or per week. And also the units might differ from vitamins, to minerals to water. I’ve assumed it is per week (so directly used the earlier modified ‘QUAN’ variables) and also assumed the nutrients can be plotted on the same scale. Because of these assumptions, some values come out as -ve which obviously don’t make sense in the real world (Food item cannot have negative amount of water).
- α_1 * (Sandwich_quantity) + α_2 * (Eggs) + …. α_{54} * (Other_Noodles) = Water_Quantity. (Here α_1 denotes how much water per sandwich, α_2 denotes how much water per egg and so on...) 54 such equations for various nutrients
- In spite of this, the results do give insights. In the sample plot (has only 10 food items for better visibility) Egg has high amount of Nitrogen (as expected), cold cereal has good vitamin D & E content, sliced cheese has low amount of nitrogen and water.
- Maybe better food_items can be chosen instead of randomly selecting the first 54 ones. This might lead to better insights. Also, the plot picked the first 10 food_items instead of all.

It uses Food questionnaire dataset: Moodle link

References:

1. Support, Confidence, Lift
2. Pearson Coefficient
3. Bokeh
4. Scikit-Learn
5. Numpy