Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 1

Due: January 18, 2023(100 points)

YOUR NAME
YOUR GIT USERNAME
YOUR E-MAIL

Some Coding Practice

In subsequent lectures, you'll learn about frequent item sets, where relationships between items are learned by observing how often they co-occur in a set of data. This information is useful for making recommendations in a rule based manner. Before looking at frequent item sets, it is worth understanding the space of all possible sets and get a sense for how quickly the number of sets with unique items grows.

Suppose that we've received only a hundred records of items bought by customers at a market. Each line in the file represents the items an individual customer bought, i.e. their basket. For example, consider the following rows.

```
Customer 1: ham, cheese, bread
Customer 2: dates, bananas
Customer 3: celery, chocolate bars
```

Customer 1 has a basket of ham, cheese, and bread. Customer 2 has a basket of dates and bananas. Customer 3 has a basket of celery and chocolate bars. Each of these records is the receipt of a given customer, identifying what they bought.

Please answer the following:

1. The *cardinality* of a set or collection of items is the number of unique items in that collection. Write a function called `cardinality_items` that takes a `.csv` text string file as input, where the format is as the above, and calculates the cardinality of the set of all the grocery items in any given dataset. What is the cardinality in "basket_data.csv"?

2. Taking any `.csv` file as a sample of a larger dataset, we'd occasionally like to understand the space of all possible subsets comprised of unique items. If there are N unique items (i.e., the cardinality of the entire dataset is N), how many sets with unique items can there possibly be? (Ignore the null set.) NOTE: I only expect the formula, and there is no code associated with this question.

3. Write a module called `all_itemsets()` with the following input/output:
   a) Input: $filename$ = the `.csv` text string file, where the format is as the above.
   b) Output: $L = [S_1, S_2, \cdots S_M]$, which is a list of all possible sets of with unique items N

4. Let's take the small sample `.csv` provided as reflective of the distribution of the receipts writ large. So, for example, if the set $S = \{\text{bread}, \text{oatmeal}\}$ occurs twice in a dataset with 100 records, then the probability of item set $\{\text{bread}, \text{oatmeal}\}$ occurring is 0.02. Write a module called `prob_S` with the following input/output:
   a) Input:
   S = the set in question
   D = the entire Dataset (which if it's in memory, Python will pass by reference). In this case, D can be a list of lists or a list of sets:
      - `[ [A, B], [A, C], [C, D] , ... ]`
      - `[ {A, B}, {A, C}, {C, D} , ... ]`
   b) Output: $P(S)$ = the probability that S occurs

Examining Our First Dataset

One of the most famous challenges in data science and machine learning is Netflix's Grand Prize Challenge, where Netflix held an open competition for the best algorithm to predict user ratings for films. The grand prize was $1,000,000 and was won by BellKor's Pragmatic Chaos team. This is the dataset that was used in that competition.

- https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

In this exercise, we're going to do a bit of exploring in the Netflix Data. Start by downloading the data. If all worked out well, you should have the files in Fig. **??**. The Kaggle dataset is close to 700MB large, and may take a long time to download. Do *not* include this data in your Docker container, but rather, mount the folder with the data.

Data Verification and Analysis

Data integrity tends to be a problem in large scale processing, especially if there is little to no support. Therefore, it's important to verify the quality of the file download.

5. A large part of machine learning and data science is about getting data in the right format. Verify that the schema is the same as the Kaggle Dataset's description. Add screenshots to your assignment.

Let's answer the following questions in your writeup:

6. How many total records are there overall?
7. Can you plot the distribution of star ratings over users and time? The granularity of the sliding window is at your discretion. Are there any trends?
8. What percentage of the films have gotten *more* popular over time?
9. How many films have been re-released? How do you know?
10. What other information might we try to extract to better understand the data? For the questions that you may come up with (especially any time series data), make sure you back up your assertions with plots. Go ahead and play around with the data, and explore.
11. What are some interesting problems that we might solve? (No need to actually solve them!)

1 Grading Criterion

A significant portion of the grading rubric is the presentation of your report. We'll review:

1. the answers to questions.
2. your code and its legibility
3. the clarity of your write-up, including
   a) pipeline and code decisions,
   b) perspectives on the solution,
   c) and algorithmic rationale.