Quantcast Summer Internship 2024 Report (Yash Vekaria)

Purpose: of this report is to summarize my approach to solve the assigned coding assignment.

Approach: My approach to extract the most active cookie corresponding to the queried date has been divided into the following steps:

1. Performing Tests to validate the user input
   I perform tests under three broad categories to test the user input:

   - *Testing cookie log file name input by the user – I perform the following checks*:
     - Length of the filename should be non-zero
     - Extension/Type of the file should be ".csv"
     - Existence of the file in the current directory

   - *Performing Tests to validate the date input by the user*:
     - Length of the date string should be non-zero
     - Length of the date string should be exactly 10 as per the instructions
     - Input should be a valid date in YYYY-MM-DD format
     - The date should not be older than the year 1994. This is because cookies were introduced on the web for the first time in 1994. So, technically, there should be no logs dated before this year.
     - The queried date should not be any future date as cookies cannot be available in the log file for any future date. (I do not check if the date in the logfile is a future date or not as that case is already eliminated by imposing a test on the input, however it can be easily done).

   - *Testing the log file contents*:
     - File content should not be empty
     - Most of the corner case scenarios are handled when parsing the log file so as it makes more sense to carry such tests at runtime. Hence, these cases are not included as part of testing but rather checked while parsing.

2. Reading the cookie log file

3. Process the cookie log file:
   This step involves the following sub-steps:

   - Preprocessing each line of the logfile to remove any whitespaces in the beginning or end of the cookie string or datetime string.
   - Ignoring a specific log file entry under the following scenarios:
     - If cookie string or datetime string has whitespace inside the string
     - If either the cookie string or datetime string is empty
     - If the cookie string is non-alphanumeric

4. Create cookie map:
   I create a dictionary of dictionary type mapping of date present in the log file to all the cookies associated with each date. Each cookie key has a corresponding value tracking the number of occurrences of the cookie value on that particular date.

5. Sorting the sub-dictionary for the queried date:
   Finally, the sub-dictionary mapping for the queried date (that contains all cookies observed on that date as keys and their frequencies as values) is extracted and sorted based on the values in descending order.

6. Printing the most active cookies:
   Finally all the cookies that occur with the same frequency as the highest frequency are printed on the console output.

Coding Style: I have followed the modular approach of writing the production grade code following appropriate naming conventions, usage of commenting for different parts of the code, and making the code reusable (if needed) by separating out different functionalities. All the test cases to validate the input are integrated in the code itself under a separate class rather than having a separate file. The runtime based scenarios are tested and handled within the code itself via exception handling and additional conditions rather than having separate test suites. Using pandas could have made the solution even more simpler. However, it has not been used as it was disallowed in the instructions. Naming style is as follows:

- Class Names:
  - CustomError()
  - MyTestSuite()
  - ProcessCookieLogfile()
- Function Names and variables:
  - These are written in lower cases with interpretable and intuitive names and multiple words are separated by underscores.

Limitations:

- One of the limitations of my approach is usage of escape characters. Since libraries like Pandas were discouraged to use, I read the csv file using the general file input/reading method. However, if “comma” is present as part of the cookie string, such entries will be ignored. CSV parsing libraries automatically escape such characters to avoid this problem.
- In addition to the above, my code has a check to ensure that the cookie string is an alphanumeric string based on the manual observation from the provided sample log file. Hence, if this is not the case, the code will not capture those other non-alphanumeric cases.

Steps to execute the code:

1. Clone the repository on your local
2. Ensure that the log file exists in the same directory as the “most_active_cookie” file.
3. Provide the name of the log file as a positional command line argument and the date of query with -d flag.
4. Finally run the code as mentioned in the instructions document provided by Quantcast.