ECOR 1042

Project Report

Manipulating CSV Files

Submitted by

TEAM T028

Omar Top, 101150311

Connor Faucher, 101235048

Xuan Nguyen, 101228417

John Coronado, 101230184

Date: 12/10/2021

Carleton University,
Faculty of Engineering and Design

1 CONTENT

2 THE PROBLEM STATEMENT

Every computer user has encountered the problem of searching through an unorganized dataset. Searching for a specific data in a list or table would be difficult and time-consuming if the entire dataset was randomized and unordered.

With an interactive user interface program, a user can select a command to be applied to the dataset by entering text commands.[1] Companies and office workers will find purpose in using a software program that helps them easily search for a particular item in a dataset or sort the dataset in a variety of ways.

	A	B	C	D	E	F	G	H
1		title	author	rating	publisher	page_cour	genres	language
2	1	Antiques Roadkill: A Trash 'n' Treasures Mystery	Barbara Allan	3.3	Kensington Publishing Corj	288	Fiction	English
3	2	The Painted Man (The Demon Cycle, Book 1)	Peter V. Brett	4.5	HarperCollins UK	544	Fiction	English
4	3	Edgedancer: From the Stormlight Archive	Brandon Sanderson	4.8	Tor Books	226	Fiction	English
5	4	Sword of Destiny: Witcher 2: Tales of the Witcher	Andrzej Sapkowski	4.8	Hachette UK	400	Fiction	English
6	5	Deadpool Kills the Marvel Universe	Cullen Bunn	4.2	Marvel Entertainment	96	Comics	English
7	6	After Anna	Alex Lake	4.1	HarperCollins UK	416	Fiction	English
8	7	How To Win Friends and Influence People	Dale Carnegie	4.3	Simon and Schuster	320	Economics	English
9	8	Little Girl Lost: A Lucy Black Thriller	Brian McGilloway	4	Harper Collins	336	Fiction	English
10	9	The Guardians: The explosive new thriller from international bestseller J	John Grisham		Hachette UK	384	Fiction	English
11	10	Marketing (The Brian Tracy Success Library)	Brian Tracy		AMACOM	112	Economics	English

Figure 1: unsorted Comma Separated Values (CSV) of Google Books dataset (acquired from Google Books store using Google API) [2]

In the text-based user interface program, the user loads a CSV file and then presses on certain keys at a time to perform the desired output.[1]

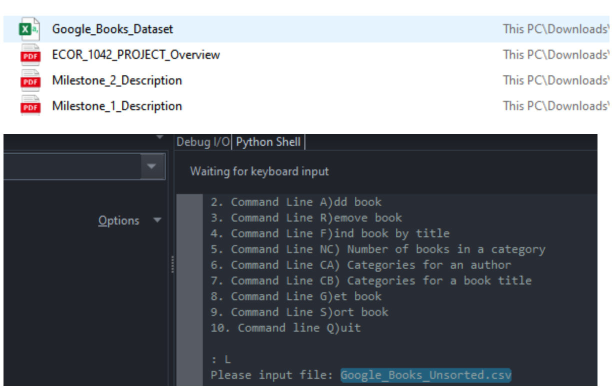

Figure 2: screenshot of text user interface of dataset analyzer program (in progress: loading file)

3 THE PROJECT GOAL

The goal of the project is to implement an interface that allows a user to provide a CSV file containing the data of multiple books.[1] From there, the user may configure and manipulate the given dataset using a variety of commands that the team creates through a series of modules containing various functions, such as sorting books by title and obtaining books by author.

4 THE PROJECT DESIGN

The program consists of a basic user interface as well as multiple modules within the interface that contain multiple functions within them. The modules include:

- "load_data.py"[2] which contains a function to read a csv file of books and organize them into a master dictionary which separates the books into lists based on their respective categories/genres.
- "search_modify_dataset.py"[2] which contains functions to modify a dictionary that is created from the function from "T028_P1_load_dataset.py" as well as functions to search for the information regarding books from the created dictionary.
- "csv_sorting.py"[1] which contains various functions that create lists of the books within the master dictionary, that are sorted/ranked by a particular factor.
- "booksUI.py"[1] is the primary file/program which contains the modules above as well as the user interface and its functions.

The user interface prompts the user of the program to select one of ten commands, all of which are displayed in the shell. The ten commands within the interface include:

1) The first command selected should be "L)oad file" which further prompts the user to input a csv file containing a list of books which will be read by the program and organize the books into a master dictionary arranged by category/genre.
2) "A)dd book", prompts the user for a book's information and calls a function that adds that book into the master dictionary.
3) "R)emove book" prompts the user for a book's information and calls a function that removes that book from the master dictionary if it was already within it.
4) "F)ind book by title" prompts the user for a book's title and searches the master dictionary to find it. The function returns "True" if the book is found and "False" if it is not found.
5) "NC) Number of books in a category" prompts the user for a category/genre and calls a function that returns the number of books from the master dictionary in that category.
6) "CA) Categories for a book title" prompts the user for an author's name and calls a function that returns a list of the categories/genres in which they have published work.

7) “CB) Categories for a book title” prompts the user for a book title within the master dictionary and calls a function that returns a list of the categories the book is a part of.
8) “G)et book” prompts the user for a specific subject of information. The user is then prompted for an example of that piece of information and calls a function which returns the books (or categories in one case) that are included in that piece of information. The subjects of information include:
    a) “R)ate”
    b) “A)uthor”
    c) “P)ublisher”
    d) “C)ategory”
    e) “CT) Category and Title”
    f) “CR) Category and Rate”
9) “S)ort book” prompts the user for a category of information and will sort all of the books of the master dictionary based on that category of information. The categories of information include:
    a) “T)itle” alphabetically.
    b) “R)ate” will further prompt the user to choose ascending or descending.
    c) “P)ublisher” alphabetically.
    d) “C)ategory” alphabetically.
    e) “PA)geCount” in ascending order.
10) “Q)uit” allows the user to quit and end the program.

5 THE PROJECT PROCESS

The program is developed incrementally using the guidelines set out in the Milestone descriptions. All Python deliverables are split into four separate cases consisting of function development and/or function testing. Each case is delegated to an individual team member and work is done simultaneously to ensure deadlines are met. For non-Python deliverables such as PDFs, the work is divided through team discussion and completed at an appropriate pace. All deliverables are reviewed by the team, and subsequently organized and submitted by the team leader.

Milestone 1 P1[2] is the first of five sets of tasks, and involves team introduction and project overview. After completing the team contract and case delegation, each team member develops a function that reads a given ‘.csv’ file and loads the data in a specified format. The ‘book_category_dictionary_list’ function is reviewed by the team, renamed as ‘load_dataset’, and submitted as the first module, ‘T028_P1_load_data.py’. Milestone 1 P2[2] involves creating 12 functionalities that manipulate the loaded dataset. When all cases are completed, team members are assigned each other's functions to develop testing modules for. All functions and tests are packaged into two separate modules, ‘T028_P2_search_modify_dataset.py’ and ‘T028_P2_search_modify_dataset_tests.py’ respectively, then submitted as team code.

For Milestone 2 P3,[1] the team develops sorting functions, which organize the loaded dataset into a sorted format. After completing the functions, team members are once again assigned each other's functions for testing, as outlined by the P3 cases. All functions and tests are packaged into two separate modules, ‘T028_P3_sorting.py’ and ‘T028_P3_sorting_test.py’ respectively. The modules are then refactored using the task guidelines and submitted as team code. Milestone 2 P4[1] involves using the functions from each previous module to build the interactive user interface outlined in the Project Goal. The separate cases are combined, split into functions, and refactored for consistency. The refactored code is submitted as the ‘T028_P4_booksUI.py’ module, concluding the main portion of the code.

The project is finalized according to Milestone 3 P5 guidelines.[3] Tasks do not need to be delegated, all team members can contribute to each task. The original ‘T028_P1_load_data.py’ module is modified to remove duplication and submitted as ‘T028_P5_load_dataset.py’. Each function module is modified to incorporate the P5 team code. The modules are subsequently reviewed, polished, and packaged into a project folder, ‘T028_data_analyzer’. After submission, a ‘README.md’ file is created for documentation and submitted alongside a final report.

6 TEAM CONTRIBUTIONS

Problem Statement: Xuan Nguyen
Project Goal: John Coronado
Project Design: Connor Faucher
Project Process: Omar Top

7 REFERENCES

[1] C. Ruiz-Martin. *ECOR 1042 – Course Project Milestone 2*. (2021). Accessed: December 10, 2021. [Online]. Available: https://brightspace.carleton.ca/d2l/le/content/90081/viewContent/2337242/View

[2] C. Ruiz-Martin. *ECOR 1042 – Course Project Milestone 1*. (2021). Accessed: December 10, 2021. [Online]. Available: https://brightspace.carleton.ca/d2l/le/content/90081/viewContent/2298655/View

[3] C. Ruiz-Martin. *ECOR 1042 – Course Project Milestone 3*. (2021). Accessed: December 10, 2021. [Online]. Available: https://brightspace.carleton.ca/d2l/le/content/90081/viewContent/2356636/View