ECOR 1042

Project Report

A Data File Analyzer Software

Submitted by

TEAM T143

SAMUEL MAJOK, 101249411

STEPHEN BOLO, 101201546

ANUSHA JAIN, 101233115

UCHENNA OBIKWELU ,101241887

April 11, 2022

Carleton University,
Faculty of Engineering and Design

1.CONTENTS

2 The Problem Statement: 2

3 The Project Goal 3

4 The Project Design: 3

5 The Project Process..... 4

6 Team Contributions 5

7 References..... 5

2 THE PROBLEM STATEMENT

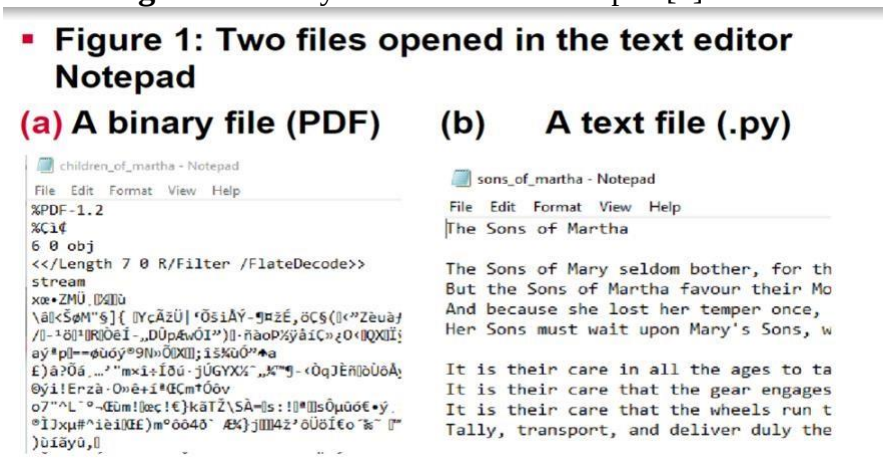
On a computer, there are generally two types of files: binary files and text files. Both contain data, but they differ in how they store the information. Binary files store their data in a digital format that is compact: as such, binary files save space, but their contents cannot be easily viewed by a human. In contrast, text files store their data in the ASCII format [4]; text files take more space to contain the same amount of data, but their contents are easily viewed by a Human.[5]

In a case where we have a large csv file that requires to be opened and specific information extracted from it, we would need to open and read the file. Reading from a CSV file is done using the reader object. The CSV file is opened as a text file with Python's built-in open() function, which returns a file object as seen in Figure 1 below. This is then passed to the reader, which does the heavy lifting. Therefore, rather than dealing with a list of individual String elements, CSV data directly into a dictionary.

Figure 1. [6]

```
file=open('google_books_dataset.csv', 'r')
l0,l1,l2,l3,l4,l5,l6=[],[],[],[],[],[],[]
for line in file:
    listDetails = line.strip().split(',')
    l0.append(listDetails[0])
    l1.append(listDetails[1])
    l2.append(listDetails[2])
    l3.append(listDetails[3])
    l4.append(listDetails[4])
    l5.append(listDetails[5])
    l6.append(listDetails[6])
```

Figure 2. Binary and text files on notepad [5].



3 THE PROJECT GOAL

The goal of this project is to build a program that implements user inputs and provides an output based on a given data set. The commands that the program will do are of the following: Add or remove books, sort the books by rate, by title, by author, by publisher and by categories(all in alphabetical order), get books by their title, within a range of a rating, by author , by publisher, or by category.

4 THE PROJECT DESIGN

This project is made up of four major functional modules. They are;

1. **T143_P5_load_data.py**[3]

The most important module out of the three others. It is a pro version of the T143_P1_load_data.py, a practice module we created as a team. It contains **one** function that reads the csv file: google_books_dataset.csv. It contains "seven headers": the book's 'title', 'author', 'rating', 'publisher', 'pages', 'category' and 'language' [3],[1].

The function converts the csv file into a dictionary with 'category' as the dictionary keys. The value of the dictionary is a list that contains 'dictionaries' as its item. The keys for these 'dictionaries' are the other 6 csv file headers and its value is, its corresponding item for the key. To do this, the function opens the file, reads its content, and iterates through each line, sorting it into a dictionary using loops and conditions [1].

The pro version of the load_data module (T143_P5_load_data.py) is very similar to its alpha version. The only difference is that the function removes duplicate entries of books that are in the same category with the same title, author, rating, publisher, pages, and language [3],[1].

2. T143_P2_add_remove_search_dataset.py[2] This

module contains eight functions:

The add_book adds a book to the dictionary and verifies if the book has been added or not. It also returns the updated dictionary. The return_book removes a book given the 'dictionary', its 'title' and its 'author' as parameters. It verifies whether the book has been removed or not and returns the dictionary [1].

The 'get_books_by …' functions take 'the dictionary' containing the data and its corresponding 'header' as parameters. The get_books_by_category function returns the number of books in that category. The get_books_by_rate returns the number of books for a given rate within a range. The get_books_by_title returns; True if the title exists in the dictionary; and False if otherwise. The get_books_by_author returns the number of books written by the author .The get_books_by_publisher function returns the number of books published by the given publisher.

Finally, the get_all_categories_for_book_title function has ‘ the dictionary’ where the data is stored and ‘a book title’ as parameters. It returns the number of categories associated with the given title [1].

3. T143_P3_sorting_fun.py[2]

This module contains four functions: sort_books_title, sort_books_publisher, sort_books_ author and sort_books_ ascending_rate. These functions take ‘the dictionary’ where the data is stored as a parameter. It converts the dictionary into a list containing dictionaries of book data as items. It then sorts the books in the required way [2].

4. T143_P4_booksUI.py[2]

This module is a text-based user interface that prompts the user to enter a command. It has the three other modules as imports. Its commands are 1.**L**)oad data, 2.**A**)dd book, 3. **R**)emove book, 4.**G**)et books and its sub commands: **T**)itle **R**)ate **A**)uthor **P**)ublisher **C**)ategory 5.**GCT**)Get all Categories for book Title 6.**S**)ort books and its subcommands: **T**)itle **R**)ate **P**)ublisher **A**)uthor 7.**Q**)uit. Depending on the user’s command input, the user interface calls the appropriate function. This user interface also works if the commands are imputed in lowercase [2].

These FOUR modules are contained in the zip file; ‘T143_data_analyzer.zip’ [3].

5 THE PROJECT PROCESS

The project started with the formation of the team and mutually signing and agreeing with the Conditions laid in the team contract. All communications took place through the Discord server.

For Milestone 1-lab 1, all the team members watched the lecture videos and took notes from it that will be helpful in doing the lab. Further, everyone watched the instructional video that explained the project and read the description. All the functions used in the lab followed the Function Design Recipe. In the first case, a ‘load_data_set’ file was created which was used throughout the project. In lab 2, more functions were created. Each member had to create 2 functions and test the other member’s functions. The functions that were created were the**.** T143_P2_add_remove_search_dataset.py functions. For testing these functions, automated testing was used, where the testing functions tested the returned values against the expected values. For the final team code for this milestone, all the functions were combined into one file and the file name was T143_P2_add_remove_search_dataset.py.[1]

For Milestone 2 - lab 3, each member had to develop a sorting function and test the other member’s function. The 4 sorting functions were sort_books_title, sort_books_publisher, sort_books_author, sort_books_ascending_rate.The final code was combining all the sorting functions into one file. For the lab - 4, each member had to develop a text-based interactive user interface for the functions previously developed. The final team code for this milestone was to combine all the user interface codes into one file. The first ITP metrics peer feedback was completed.[2]

For Milestone 3, the first task as a team was to remove the duplicate entries from the dictionary and after that combine all the codes from previous milestones into one zip file. Further, the team developed a README.md file that introduced the project to a user and told them how the project functions. Finally, the project report was written and edited by all the members. After that, the second ITP metrics peer feedback was completed.[3]

6 TEAM CONTRIBUTIONS

Samuel Majok	The Problem Statement, Team Contributions, References
Stephen Bolo	The Project Goals, References
Uchenna Obikwelu	The Project Design, References
Anusha Jain	The Project Process, References

7 REFERENCES

[1] J. Hosseinkhani and C. Ruiz Martin, "Milestone_1_Description,Brightspace.carleton.ca. 2022. [online].Available: https://brightspace.carleton.ca/d2l/le/content/124606/viewContent/2561458/View [Accessed: 10 April2022].

[2] J. Hosseinkhani and C. Ruiz Martin, "Milestone_2_Description,Brightspace.carleton.ca. 2022. [online].Available: https://brightspace.carleton.ca/d2l/le/content/124606/viewContent/2543455/View [Accessed: 10 April 2022].

[3] J. Hosseinkhani and C. Ruiz Martin, "Milestone_3_Description,Brightspace.carleton.ca. 2022. [online].Available: https://brightspace.carleton.ca/d2l/le/content/124606/viewContent/2543473/View [Accessed: 10 April 2022].

[4] "ASCII table," ASCII Table - ASCII Character Codes, HTML, Octal, Hex, Decimal. [Online]. Available: https://www.asciitable.com/. [Accessed: 10 April 2022].

[5] J. Hosseinkhani and C. Ruiz Martin, "Lecture 3: Iterative Development and Text Processing Brightspace.carleton.ca. 2022. [online] Available: https://brightspace.carleton.ca/d2l/le/content/124606/viewContent/2543410/View. [Accessed: 10 April 2022].

[6] Team T143, 'P5-T2 Team Code, T143_data_analyzer.zip' Brightspace.carleton.ca. 2022. [online] Available: https://brightspace.carleton.ca/d2l/le/content/124606/viewContent/2543467/View. [Accessed: 10 April 2022].