Assignment Coversheet

Student No:	MSAI7/20.1/0011
Student Name:	Sean Sciberras
Programme Title:	M.Sc. in Artificial Intelligence
Module No & Title:	9 - Decision Support Systems and Expert Systems
Name of lecturer:	Gerard Said Pullicino
Due Date:	15 March 2023
Word Count:	3,068

Assignment Question:

Create a book recommendation system using CLIPS

You must keep a softcopy of your coursework

Student Declaration

I certify that the attached assignment is my own work. Information drawn from different sources has been appropriately referenced and fully acknowledged in the relevant section. Referencing throughout this assignment follow Harvard Referencing Style.

Sean Sciberras	15 March 2023
Student Name/ Signature	Submission Date

Assignment must be submitted electronically on CANVAS Platform in the relevant section. Kindly type your name signature space above, however a signed copy must be produced on request.

Introduction

Decision Support Systems (DSS) are tools designed to assist complex decision-making, especially where a user lacks the processing speed and power to deal with large amounts of data and complex analysis. Recommendation systems are automated systems that filter information that a decision maker may need in order to maximise the likelihood that the user will be satisfied with the selection's outcome. (Stohr and Viswanathan 1998). Schmalhofer (2001) elaborates on how Expert Systems (ES) are computer systems that demonstrate a level of intelligence comparable to that of human experts. They typically consist of a knowledge base, a search or inference system, a knowledge acquisition system, and a user interface or communication system. Knowledge systems tackle complex problems in the real world by the application of inference procedures to explicitly specified knowledge.

Book recommendation systems would employ machine learning and data mining to assess user data, such as reading history and ratings, and based on their interests, would provide users with book suggestions that they are most likely to appreciate. Book recommendation systems may be categorised as a type of DSS because they help users make informed judgements about what to read next. At the same time, they use artificial intelligence to duplicate the knowledge and decision-making abilities of human experts in book recommendations, so book recommendation systems can also be considered ES. One can argue that book recommendation systems can be categorised as a form of an intelligent system that combines the capabilities of DSS and ES to give users customized recommendations.

According to Nikzad–Khasmakhi, Balafar, and Reza Feizi–Derakhshi (2019) and Kurmashov, Latuta and Nussipbekov (2015), recommendation systems may be categorised as collaborative recommendation systems, content-based filtering systems, or hybrid systems. Collaborative recommendation systems select and present products based on similarity measurements. Collaborative recommendation systems can be further classified as user–user collaborative filtering, item–item collaborative filtering, or a combination of both. The technique of user–user collaborative filtering generates recommendations based on the contributions of other users in the same community, whereas item–item collaborative filtering focuses on locating related objects. On the other hand, content-based recommendation systems filter items based on the user's previous rating. While hybrid recommendation systems attempt to combine and integrate the different recommendation methodologies.

Introduction to the Jupyter notebook

The Appendix at the end of the document shows a flow chart of the processes discussed below, highlighting how the recommendation system makes use of CLIPS facts and rules to recommend further book titles to the user. The notebook is organised in 4 sections:

- Section 1: includes some **preliminary steps** like uploading libraries to be used in the notebook and loading the dataset.
- Section 2: includes **dataset exploration** and **preparation** so it can be used by the recommender system.
- Section 3: simulates a **user selection of ten books**. A random category out of 192 book categories is selected and ten random book titles within that category are chosen.
- Section 4: is the **recommender system** that asserts book facts and runs CLIPS rules to identify a common category between the selected books. The category may actually vary from the one in Section 3 but is usually quite similar to that category. The rules lead to more asserted facts until the user is presented with a ranked recommended list of books that should share similar book categories as the books selected in Section 3.

The notebook and dataset were also made available on an online platform: https://github.com/Sean-from-Malta/Module-9

Provided Dataset

A Github dataset called “Amazon popular books dataset” was provided at https://github.com/luminati-io/Amazon-popular-books-dataset

It is stated that the dataset contains 2,269 most reviewed and best-selling books on Amazon, with each book having at least 10,000 reader reviews. The Amazon dataset was extracted in the second week of May 2022 using the Bright Data Amazon Scraper.

Amazon.com (2023a) boasts as a one-stop-shop for book lovers that it has a vast selection of book titles to suit the preferences of any reader in any category from “literature, cookbooks, mystery, comics, romance, memoirs, history, science fiction and children's books” with a variety of authors and book series. Observing its website, Amazon seems to group books under 32 main categories.

The dataset was uploaded in the notebook as a dataframe named amazon_df and contained 2,269 rows (representing different book titles) with 40 columns or features. print (amazon_df.dtypes) can be used to obtain column names and relative data types. Another dataframe named df was created retaining just the 'rating', 'title', 'best_sellers_rank' and 'categories' columns.

The main Amazon book categories

The 'best_sellers_rank' and 'categories' columns were reviewed to identify book titles within the dataset falling under the 32 main Amazon book categories. Book categories were found not to be evenly distributed and most of the books were classified under more than one category:

Amazon Main Categories	**No of Books**
Calendars:	1
Engineering & Transportation:	8
Computers & Technology:	13
Test Preparation:	15
Sports & Outdoors:	26
Parenting & Relationships:	29
Cookbooks, Food & Wine:	33
Education & Teaching:	35
Law:	36
LGBTQ+:	38
Arts & Photography:	44
Comics & Graphic Novels:	45
Crafts, Hobbies & Home:	55
Science & Math:	57
Travel:	59
Christian Books & Bibles:	64
Humor & Entertainment:	74
Medical:	88
Business & Money:	110
Religion & Spirituality:	124
Self-Help:	134
Health, Fitness & Dieting:	147
Reference:	151
Teen & Young Adult:	161
History:	176
Politics & Social Sciences:	180
Biographies:	190
Romance:	241
Science Fiction & Fantasy:	355
Children:	358
Mystery, Thriller & Suspense:	479
Literature & Fiction:	1,089

No of Categories per Book	**No of Books**
1	765
2	946
3	360
4	134
5	47
6	12
7	5
Total	**2,269**

Reedsy (2023) and Munton (2022) explain that books can be categorised broadly as either fiction or nonfiction, but can be classified further with Reedsy using 107 genres and Amazon employing thousands of distinct categories and subcategories. Amazon categories appear to be based on BISAC, also known as the Book Industry Standards and Communications code (Amazon, 2023e). Classifications can be extremely specific, for instance, FIC014050: Fiction/Historical/20th Century/World War II. (BISAC, 2023) The majority of libraries organise their books using the Dewey Decimal System, which includes well-defined categories and hierarchies as well as a dense network of relationships between topics. (Online Computer Library Center in 2003; William O. Schaefer Elementary School in 2018)

Additionally, Wikipedia categories are also an excellent source of ideas for grouping and organising categories[1]. The assignment required the use of categories in the recommendation system. After evaluating the various approaches, it was decided to make the best use of CLIPS capabilities and utilise all categories available in the "categories" and "best sellers rank" columns of the dataset without any hierarchies or groupings. So for instance, if a book was classified under FIC014050, the system would classify the book under four distinct categories, "Fiction," "Historical," "20th Century," and "World War II".

The Amazon rating system

Amazon's 5-star rating system is an integral component of the platform's overall review system. It allows customers to rate products on a scale from one to five stars, with five stars representing the highest rating and one star representing the lowest. This system is utilised to provide an overall picture of a product's quality and customer satisfaction, as well as to assist other customers in making educated purchasing decisions. (Amazon.com, 2023(b)(c)(d) and Skubana, 2018).

When calculating a product's overall rating, the system takes into account a variety of factors such as the total number of ratings, the average rating, the proportion of positive ratings, the age of the rating and whether the rating is the result of a verified purchase. The verified purchase status is an integral component of Amazon's 5-star rating system, as customers must have spent

[1] See: https://en.wikipedia.org/wiki/Wikipedia:Contents/Indices
https://en.wikipedia.org/wiki/Wikipedia:Contents/Categories
https://en.wikipedia.org/wiki/Wikipedia:Contents/Glossaries
https://en.wikipedia.org/wiki/Wikipedia:Contents/Lists
https://en.wikipedia.org/wiki/Wikipedia:Contents/Outlines

at least $50 on their account in order to submit a review and purchased the item on Amazon for at least 80 percent of its original price. This helps to ensure the authenticity and precision of ratings and reviews. The rating feature appears to be a good basis to rank the algorithm results. (Amazon.com, 2023(b)(c)(d) and Skubana, 2018).

The code df['rating'] = [re.sub(' out of 5 stars', '', x) for x in df['rating']] can be used to remove the string “out of 5 stars” from the column “rating” in our dataframe to transform the variable in a number which can be used further by the algorithm. A closer look at unique values within the column using the code df.rating.unique() showed that book ratings ranged between 3.4 and 4.9.

Bias

Chen et al. 2020 explain how in recommendation systems there is a cycle of data collection, analysis, recommendation, and user feedback, with each stage having the potential to contribute to biases that influence the recommendations made to users. One of the major issues is that it can result in a lack of diversity in user recommendations. As popular items become more popular and unpopular items become less popular, users may only be exposed to a limited range of items that reinforce their existing beliefs and opinions. Top items typically receive more traffic, which increases their ranking prominence and the amount of traffic they receive, resulting in a situation where the “rich get richer”.

They explain how research has demonstrated that users tend to behave similarly to others in a group, even if doing so goes against their own judgement, thereby indicating that user feedback does not always reflect the user's true preference. Also, users are usually free to choose which items to rate, so actual ratings may not be a representative sample of all ratings. Moreover, because users are only exposed to a subset of the population, less popular interactions do not necessarily indicate negative preferences. They refer to this phenomenon as the feedback loop, in which biases in recommender systems can be self-reinforced and even amplified over time.

The recommendation system will need to work with a provided list of ten book titles and suggest additional book titles. When constructing a system for recommending books, it is essential to handle duplicate titles correctly to ensure accurate recommendations. Duplicates can distort the dataset leading to biased recommendations. A basic check in the dataset using the code df[df.duplicated(['title'])] revealed 668 duplicate titles in the dataset.

Even after removing these 668 titles, a deeper look revealed other duplicates. “12 Years A Slave” was not identified as being the same as “12 Years a Slave”. Nor was “A Killer's Wife (Desert Plains Book 1)” considered identical to “A Killer's Wife: Desert Plains, Book 1”. It was decided to remove such similar titles by creating a new dataframe column containing a “cleaned up” version of the title as a lowercase title, without grammatical conjunctions “and” and “or”, articles “a”, “an” and “the”, non-alphanumeric characters and the words “book” and “novel” and using it to remove duplicates.

The dataset almost certainly still contains duplicates. For instance, “21st Birthday”, “21st Birthday (Women’s Murder Club)” and “21st Birthday (Women’s Murder Club, 21)” might be duplicate book titles, but it is difficult to tell without additional information. They could easily be multiple versions or editions of the same book. However, it was decided to leave them in the dataset because examining the remaining 1,460 book titles would exceed the scope of the assignment.

The Book Recommender System

1) CLIPS

Riley (2021) explains that the C Language Integrated Production System (CLIPS) is a rule-based programming language that can be used to build an expert system and that it is a public domain software package. One of the strengths of CLIPS is its ability to handle large amounts of complex data and to make complex decisions based on that data through the definition of facts and rules. Pathapati (2023) explains how a CLIPS expert system uses forward chaining, that is matching a set of initial facts and using the rules to generate new facts, which can, in turn, trigger more rules until eventually, the system would in our case recommend additional books. In the notebook, the code establishes a CLIPS environment using the code `env = Environment()` and includes a custom logging router `PrintLoggingRouter()` that appends generated messages to a list named `favourite_books` using the code:

```
class PrintLoggingRouter(clips.LoggingRouter):
    def write(self, name, msg):
        text = msg
        favourite_books.append(text)
        if msg.startswith('WARN:'):
            logging.warn(msg[5:])
```

```
        elif msg.startswith('ERROR:'):
            logging.error(msg[6:])
        else:
            logging.info(msg)

env.add_router(PrintLoggingRouter())
```

The list favourite_books will be used in Section 4 of the algorithm to generate the results.

2) Templates

In Section 4, the code includes a CLIPS template for "book" facts:

```
booktemplate = """
(deftemplate book
  (slot title (type STRING))
  (multislot category (type SYMBOL))
  (slot rating (type FLOAT)))
"""

env.build(booktemplate)
```

Templates define the structure of the facts that the expert system will assert. CLIPS templates specify the slots that a fact may contain and the type of data that may be stored in each slot. The "book" template in the code contains three fields: "title", "category" and "rating". The "title" slot stores the book's title as a string, the "category" slot stores one or more categories that the book belongs to as a list of symbols, and the "rating" slot stores the book's rating as a float. The "env.build()" function is then used in CLIPS to create a new "book" fact based on a template defined in the environment.

In a similar manner, the "selection" template has the same structure as the "book" template, with the exception that it is used to represent the 10 user-selected book facts. It is crucial to define templates in CLIPS so that all facts of a particular type have the same structure. There are 3 other templates in the algorithm:

- selected to assert the titles of 10 user-selected books as facts,
- favcateg to assert the most common category chosen by the recommender system for the 10 user-selected books as a fact and
- favourite to assert books within the most common category as facts.

3) Asserting Facts

In CLIPS, asserting facts is the process of adding new knowledge to the expert system which can then be developed to reason about complex problems. When a fact is asserted, it becomes part of the expert system's working memory and can be used for pattern matching, manipulation, and meaningful reasoning through the application of CLIPS rules. By asserting facts that represent the current state of the problem, the expert system is able to reason about it and make decisions accordingly. New facts can be added to the expert system's knowledge base to update its understanding of the problem. New facts can trigger rules that match those facts and infer new knowledge. This enables the expert system to derive deductions and draw conclusions based on the asserted knowledge.

The code converts each row of the DataFrame to an asserted "book" fact in the CLIPS environment. If the book title is one of the 10 user-selected books it is asserted as a "selected" fact:

```
for index, row in final_book_list.iterrows():
  fact_string = f"(book (title \"{row['title']}\") (category {' '.join(row['category'].split())}) (rating {row['rating']}))"
   env.assert_string(fact_string)

for title in selected_titles:
   fact_string = f"(selected (title \"{title}\"))"
   env.assert_string(fact_string)
```

4) CLIPS rules

The asserted facts are then used by the "defrule remove-book" rule in the code to identify books that are included in both as "book" and "selected" facts, assert "selection" facts and retract the "book" facts, leaving a complete list of facts that are either "book" or "selection" facts:

```
(defrule selected-book
   ?selected  <- (selected (title ?title))
   ?book <- (book (title ?title) (category $?category) (rating ?rating))
=>
   (assert (selection (title ?title) (category $?category) (rating ?rating)))
   (retract ?book)
   (printout t "Categories: " $?category crlf)
)
```

The Left-Hand Side (LHS) of the rule consists of those elements before the "=>" and matches the same title in "book" and "selected" facts. When a match is found, it triggers the Right-Hand Side (RHS) of the rule, that is those elements after the "=>", in this case, assert "selection" facts and retract the matched "book" fact. Categories of "selection" facts are captured by the custom logger and saved in the favourite_books list

Thus, the remaining asserted facts are either "book" facts that were not chosen or "selection" facts that have no "book" counterparts. This provides a comprehensive listing of unique facts that can be used for additional analysis and processing. The "env.build" command is used to load the rule into the CLIPS environment, followed by the "env.run" command to run the rule in the environment and look for any matching facts.

The code also uses the favourite_books list to identify most_common_categ, a book category present in the 10 user-selected books, which in turn is asserted as a favcateg fact.

5) Forward chaining

The clean_category function is used to create a new dataframe column containing only the selected favourite book category most_common_categ if this is present in the "categories" column. All dataframe rows are then asserted as new book facts but with updated categories.

The selection-book rule then matches the selection title facts with book facts and retracts the matched user-selected ten book facts.

This in turn leads to the add-to-favorites rule which matches the category in the asserted favcateg fact to the updated category of asserted book facts and asserts favourite facts with just book title and rating.

The asserted favourite facts are matched by the favourite-books rule and when the rule fires, the asserted favourite fact is appended to the favourite_books list by the custom logging router PrintLoggingRouter() similar to the selected-book rule mentioned earlier.

6) Giving the recommendation to the user

The code then iterates through favourite_books to extract the book titles and ratings and adds them to the "books" list as a tuple. The "books" list is then sorted by rating using the code:

```
books.sort(key=lambda favourite_books: favourite_books[1], reverse=True)
```

Finally, the list is then printed to give to the user the list of recommended books in descending ranking order.

Conclusion

The CLIPS platform was used to demonstrate how an expert system can be set up to carry out a cycle of asserting facts to fire up rules leading to additional asserted facts and further rules firing up. The expert system is thus capable of making book recommendations to a user based on a list of 10 titles selected by the user earlier.

The CLIPS platform was used to demonstrate how an expert system can be configured to assert facts that lead to a loop of rule triggers and further fact assertions. In the end, the expert system is capable of recommending books to a user based on a list of 10 titles that the user had selected previously.

References

Amazon.com(a). 2023. *Books at Amazon*. [online] Available at: https://www.amazon.com/books-used-books-textbooks/b/?ie=UTF8&node=283155&ref_=topnav_storetab_b [Accessed 4 March 2023]

Amazon.com(b). 2023. *Understanding Customer Reviews and Ratings*. [online] Available at: https://www.amazon.com/gp/help/customer/display.html?nodeId=G8UYX7LALQC8V9KA [Accessed 4 March 2023]

Amazon.com(c). 2023. *How Are Product Star Ratings Calculated?* [online] Available at: https://www.amazon.com/gp/help/customer/display.html?nodeId=GQUXAMY73JFRVJHE [Accessed 4 March 2023]

Amazon.com(d). 2023. *Amazon Verified Purchase Reviews*. [online] Available at: https://www.amazon.com/gp/help/customer/display.html?nodeId=G75XTB7MBMBTXP6W [Accessed 4 March 2023]

Amazon.com(e). 2023. *KDP Categories*. [online] Available at: https://kdp.amazon.com/en_US/help/topic/G200652170 [Accessed 20 February 2023]

BISAC. 2023. *BISAC Subject Headings List*. [online] Available at: https://www.bisg.org/fiction [Accessed 20 February 2023]

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M. and He X. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *Association for Computing Machinery*. **1**(1) [online] Available at: https://arxiv.org/pdf/2010.03240.pdf [Accessed 5 March 2023]

Kurmashov, N., Latuta, K. and Nussipbekov, A. 2015. *Online book recommendation system*. Twelve International Conference on Electronics Computer and Computation (ICECCO September 2015) pp 1-4 [online] Available at: https://www.researchgate.net/publication/300412849_Online_book_recommendation_system [Accessed 5 March 2023]

Munton, R. 2022. *A comprehensive guide to the different types of books*. [online] Available at: https://www.bookbeaver.co.uk/blog/different-types-of-books [Accessed 20 February 2023]

Nikzad–Khasmakhi, N., Balafar, M.A. and Reza Feizi–Derakhshi, M. 2019. The state-of-the-art in expert recommendation systems, [online] *Engineering Applications of Artificial Intelligence*, **82**(June 2019) pp 126-147 Available at: https://www.sciencedirect.com/science/article/abs/pii/S0952197619300703?via%3Dihub [Accessed 4 March 2023]

Online Computer Library Center. 2003. *Summaries*. [online] Available at: https://www.oclc.org/content/dam/oclc/dewey/resources/summaries/deweysummaries.pdf [Accessed 20 February 2023]

Pathapati, M. 2023. *Engineering the AI applications (Part 3): Use of CLIPS for Forward Chaining applications*. [online] Available at: https://www.linkedin.com/pulse/engineering-ai-applications-part-3-use-clips-forward-pathapati/

Reedsy. 2023. *The Ultimate List of Book Genres: 35 Popular Genres, Explained*. [online] Available at: https://blog.reedsy.com/book-genres/ [Accessed 20 February 2023]

Riley, G. 2021. *CLIPS Reference Manual Volume I Basic Programming Guide*. [online] Available at: https://clipsrules.net/documentation/v640/bpg640.pdf [Accessed 5 March 2023]

Schmalhofer, F. 2001. Expert Systems in Cognitive Science. [online] *International Encyclopedia of the Social & Behavioral Sciences*. pp 5128-5135. Available at: https://www.sciencedirect.com/science/article/pii/B0080430767016156 [Accessed 4 March 2023]

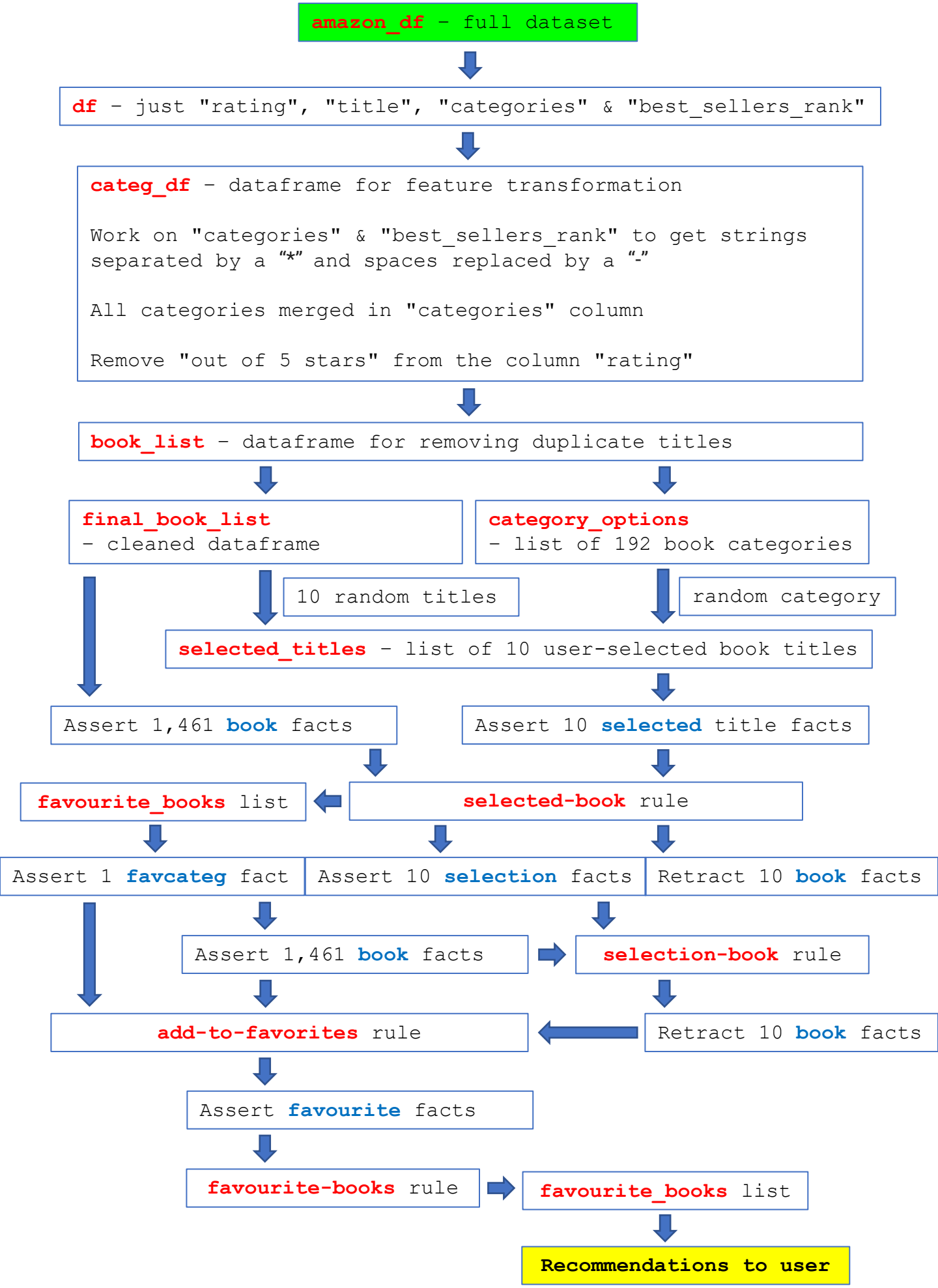
Skubana. 2018. *Everything You Need To Know About Amazon Verified Product Reviews*. [online] Available at: https://sellerengine.com/amazon-verified-product-reviews/ [Accessed 4 March 2023]

Stohr, E.A. and Viswanathan, S. 1998. *Recommendation Systems: Decision Support for the Information Economy*. [online] Available at: https://archive.nyu.edu/bitstream/2451/14303/1/IS-98-17.pdf [Accessed 4 March 2023]

William O. Schaefer Elementary School. 2018. *How Nonfiction is Organized*. [online] Available at: https://blogs.socsd.org/woslmc/2018/11/06/how-nonfiction-is-organized/ [Accessed 20 February 2023]

Appendix

Flow Chart showing the interaction between various components in the algorithm to give a recommendation of book titles to the user


amazon_df - full dataset
df - just "rating", "title", "categories" & "best_sellers_rank"
categ_df - dataframe for feature transformation
Work on "categories" & "best_sellers_rank" to get strings separated by a "*" and spaces replaced by a "-"
All categories merged in "categories" column
Remove "out of 5 stars" from the column "rating"
book_list - dataframe for removing duplicate titles
final_book_list
- cleaned dataframe
category_options
- list of 192 book categories
10 random titles
random category
selected_titles - list of 10 user-selected book titles
Assert 1,461 book facts
Assert 10 selected title facts
favourite_books list
selected-book rule
Assert 1 favcateg fact
Assert 10 selection facts
Retract 10 book facts
Assert 1,461 book facts
selection-book rule
add-to-favorites rule
Retract 10 book facts
Assert favourite facts
favourite-books rule
favourite_books list
Recommendations to user