UNIVERSITÀ
DEGLI STUDI
DI MILANO

Market Basket Analysis

Algorithms for massive data

Farnaz Sharbafi farnaz.sharbafi@studenti.unimi.it

Abstract

This report explores the application of association rule mining, specifically utilizing the FP-Growth algorithm, to analyze a dataset of book reviews. The dataset includes user reviews, book titles, and user identifiers. In this project a dual perspective on frequent itemsets is explored: (1) words used within review texts, and (2) sets of books reviewed by the same user. Association rule mining is a powerful data mining technique for uncovering meaningful patterns and co-occurrences. This study focuses on discovering frequent word combinations and identifying books that tend to be reviewed together, which can provide insights into user interests, reading behavior, and potential recommendations. By leveraging the FP-Growth algorithm within a scalable data processing environment, the analysis demonstrates how valuable patterns can be extracted from large volumes of unstructured and structured review data. These insights can support better book categorization, personalized recommendations, and understanding of reader communities.

Keywords: Association Rule Mining, Book Reviews, FP-Growth Algorithm

1. Introduction

The idea of frequent item sets is at the heart of association rule mining, a data mining technique used to find links between items in a dataset. Finding association rules, which highlight patterns of things that frequently occur together, is based on these frequent item sets. The main objective is to identify significant relationships between objects that commonly appear together in a collection of our dataset to gain insightful knowledge. Association rule mining can be applied across a variety of fields including Retail and E-commerce, healthcare, telecommunications, marketing and more for finding hidden patterns and relationships.

Ahn and Kim mentioned that association rule mining is a significant area of research within the field of data mining. It typically involves two primary tasks: first, identifying frequent itemsets, and second, deriving interesting association rules from these itemsets. The computational complexity of discovering frequent itemsets makes efficiency a critical factor in the first step. The second step, which involves the extraction of meaningful rules, requires unbiased evaluation. Apriori and FP-Growth are two examples of association rule mining algorithms that are used to identify frequent item sets and produce association rules. Iteratively creating candidate item sets and eliminating those that don't reach the minimal support level is how these algorithms operate. After identifying the frequent itemsets, we can use the concept of confidence to create association rules. Confidence measures how often items in the consequent appear in transactions that already contain the antecedent. Ari Aldino et al. discussed that by processing transaction data companies are able to find out consumer buying patterns. Based on their analysis, the Fp-Growth algorithm worked better than the Apriori algorithm because of higher rule strength and accuracy.

This analysis can benefit publishers, recommendation systems, and online retailers by identifying patterns in user preferences and reading habits. Market Basket Analysis (MBA) is a technique used to uncover relationships between items in a dataset—commonly applied in retail to understand consumer behavior. In the context of book reviews, MBA helps reveal associations between books frequently reviewed by the same user and common themes or expressions used across review texts. These associations are discovered using frequent itemset mining and association rule techniques, such as the FP-Growth algorithm.

The insights gained from this analysis can enhance personalized book recommendations, improve book categorization, and inform marketing strategies. For instance, understanding which books tend to be reviewed together could help suggest new books to readers, while analyzing co-occurring words in reviews would reveal customer opinions and preferences that can guide marketing, product improvements, or recommendation strategies. In this project, a dataset of book reviews is analyzed using the FP-Growth algorithm to detect frequent combinations of words and book sets.

2. Methodology

Apache Spark is a distributed computing system designed to dramatically improve the speed and efficiency of processing large-scale data. It has been developed in 2009 as part of a research initiative at UC Berkeley's AMPLab. Spark was created to offer a fast, iterative framework which is ideal for machine learning, interactive data analysis, and other data-intensive tasks. Apache Spark works by distributing data and computation across a cluster of machines to process large-scale datasets quickly. It divides tasks into smaller units, processes them in parallel, and combines the results. A standout feature of Spark is its in-memory caching and optimized query execution, which allows for fast analysis of big datasets. It supports multiple programming languages, including Java, Scala, Python, and R, promoting code reuse across various workloads like batch processing, real-time analytics, interactive queries, machine learning, and graph processing.

Application Programming Interfaces (APIs) are tools that allow two software systems to interact with each other by following a set of rules and guidelines. PySpark, the Python API for Apache Spark, extends these capabilities by enabling large-scale data processing within a distributed framework using Python. It also provides an interactive PySpark shell for real-time data exploration. By combining Python’s ease of use with Spark’s powerful distributed processing, PySpark allows Python developers to efficiently process and analyze data at any scale. Unlike Pandas, which struggles with large datasets, PySpark utilizes cluster computing to perform parallel processing, significantly speeding up computations and reducing processing times. Its combination of simplicity and performance makes PySpark a suitable tool for scalable, high-performance data solutions.

2.1. FP-Growth algorithm

FP-Growth is a highly efficient algorithm used in data mining for discovering frequent itemsets and association rules, making it particularly popular for market basket analysis. “FP” denotes to Frequent Patterns. In the initial stage of the FP-Growth algorithm, the main task is to calculate the frequency of individual items and identify those that appear frequently within the dataset. This step provides the foundation for the subsequent discovery of larger frequent itemsets.

The FP-Growth (Frequent Pattern Growth) algorithm extracts frequent patterns from a dataset without relying on candidate generation. It begins by counting item frequencies and filtering out those below the minimum support threshold. The remaining items are then ordered by frequency, and transactions are added sequentially to build a compact pattern structure that captures frequent relationships efficiently. The FP-Growth algorithm finds frequent patterns efficiently because it doesn’t rely on generating and testing candidates, which makes it especially useful for large datasets. When applying it to association rule mining, different measures are used to judge how meaningful the rules are. It helps how strong and reliable the connections between items really are in the data.

The effectiveness of the discovered association rules is evaluated using three key metrics:

Support: It is one of the measures of interestingness that measures how frequently an itemset appears in the dataset.

$$Support(X) = \frac{Transactions\ containing\ X}{Total\ number\ of\ transactions}$$

Confidence indicates the likelihood that an itemset Y will appear in transactions containing itemset X.

$$Confidence(X \Rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)}$$

Lift assesses the strength of an association rule by comparing the observed co-occurrence of X and Y with what would be expected if X and Y were independent.

$$Lift(X \Rightarrow Y) = \frac{Confidence(X \Rightarrow Y)}{Support(Y)}$$

3. Experimental result

The FP-Growth algorithm is implemented using PySpark’s FPGrowth module, which efficiently handles large-scale datasets, enabling robust analysis and insight extraction.

To prepare the dataset for analysis, the following steps were performed:

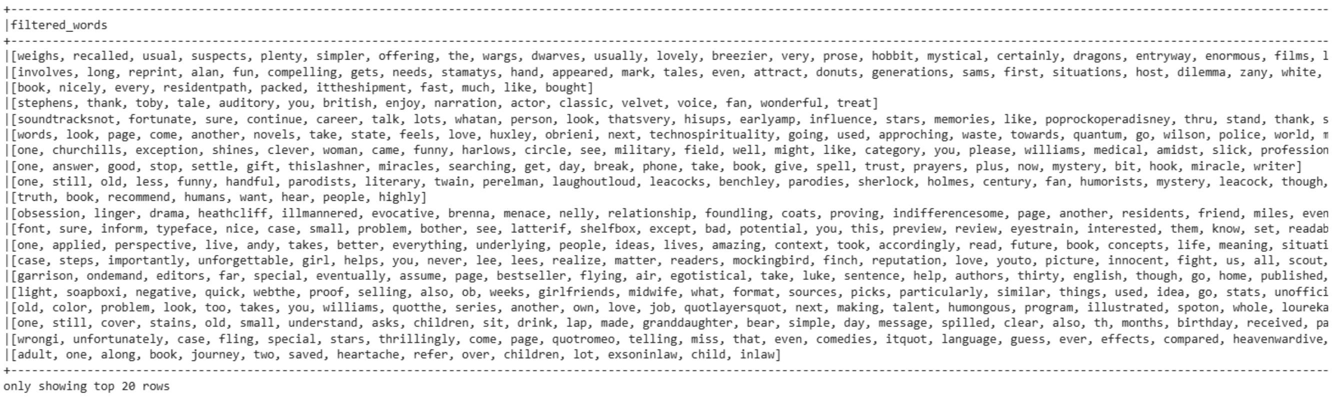
I began by uploading the necessary Kaggle credentials (kaggle.json) to Google Colab. This file contains the authentication details required to access datasets from Kaggle via the Kaggle API. After uploading the credentials, they were securely stored in the appropriate directory and file permissions were adjusted. This ensures that the API can use the credentials without exposing sensitive information. Using the Kaggle API, the *Books_rating.csv* file was downloaded directly into the working environment. The dataset was then extracted and made available for further processing.

This dataset includes 3 million book reviews. It contains ten columns, *Id, Title, Price, User_id, ProfileName, review/helpfulness, review/score, review/time, review/summary, review/text*.

There are 43 rows with null values in the 'review/text' column and 208 rows with null values in the 'title' column that dropped out before analysis. There are also 8865 duplicate rows in the dataset that deduplicated before analysis.

In the first step, a system finding frequent itemsets considering words as items is implemented. The process involves normalizing text reviews by converting text to lowercase. Tokenization was then applied to split the text into individual words, and common English stopwords (e.g., “the,” “and,” “is”) were removed. Each review was treated as a transaction, with the remaining meaningful words considered as items for frequent itemset mining. The dataset was then transformed into a format compatible with the FP-Growth algorithm. Due to the size of the dataset, 0.001 of the rows are considered for further analysis.

Below in figure 1 the processed DataFrame, cleaned and optimized is shown.



```
+----------------------------------------------------------------------------------------------------------------------------------------------------------------------
|filtered_words
+----------------------------------------------------------------------------------------------------------------------------------------------------------------------
|[weighs, recalled, usual, suspects, plenty, simpler, offering, the, wargs, dwarves, usually, lovely, breezier, very, prose, hobbit, mystical, certainly, dragons, entryway, enormous, films, l
|[involves, long, reprint, alan, fun, compelling, gets, needs, stamatys, hand, appeared, mark, tales, even, attract, donuts, generations, sams, first, situations, host, dilemma, zany, white,
|[book, nicely, every, residentpath, packed, ittheshipment, fast, much, like, bought]
|[stephens, thank, toby, tale, auditory, you, british, enjoy, narration, actor, classic, velvet, voice, fan, wonderful, treat]
|[soundtracksnot, fortunate, sure, continue, career, talk, lots, whatan, person, look, thatsvery, hisups, earlyamp, influence, stars, memories, like, poprockoperadisney, thru, stand, thank, s
|[words, look, page, come, another, novels, take, state, feels, love, huxley, obrieni, next, technospirituality, going, used, approching, waste, towards, quantum, go, wilson, police, world, m
|[one, churchills, exception, shines, clever, woman, came, funny, harlows, circle, see, military, field, well, might, like, category, you, please, williams, medical, amidst, slick, profession
|[one, answer, good, stop, settle, gift, thislashner, miracles, searching, get, day, break, phone, take, book, give, spell, trust, prayers, plus, now, mystery, bit, hook, miracle, writer]
|[one, still, old, less, funny, handful, parodists, literary, twain, perelman, laughoutloud, leacocks, benchley, parodies, sherlock, holmes, century, fan, humorists, mystery, leacock, though,
|[truth, book, recommend, humans, want, hear, people, highly]
|[obsession, linger, drama, heathcliff, illmannered, evocative, brenna, menace, nelly, relationship, foundling, coats, proving, indifferencesome, page, another, residents, friend, miles, even
|[font, sure, inform, typeface, nice, case, small, problem, bother, see, latterif, shelfbox, except, bad, potential, you, this, preview, review, eyestrain, interested, them, know, set, readab
|[one, applied, perspective, live, andy, takes, better, everything, underlying, people, ideas, lives, amazing, context, took, accordingly, read, future, book, concepts, life, meaning, situati
|[case, steps, importantly, unforgettable, girl, helps, you, never, lee, lees, realize, matter, readers, mockingbird, finch, reputation, love, youto, picture, innocent, fight, us, all, scout,
|[garrison, ondemand, editors, far, special, eventually, assume, page, bestseller, flying, air, egotistical, take, luke, sentence, help, authors, thirty, english, though, go, home, published,
|[light, soapboxi, negative, quick, webthe, proof, selling, also, ob, weeks, girlfriends, midwife, what, format, sources, picks, particularly, similar, things, used, idea, go, stats, unoffici
|[old, color, problem, look, too, takes, you, williams, quotthe, series, another, own, love, job, quotlayersquot, next, making, talent, humongous, program, illustrated, spoton, whole, loureka
|[one, still, cover, stains, old, small, understand, asks, children, sit, drink, lap, made, granddaughter, bear, simple, day, message, spilled, clear, also, th, months, birthday, received, pa
|[wrongi, unfortunately, case, fling, special, stars, thrillingly, come, page, quotromeo, telling, miss, that, even, comedies, itquot, language, guess, ever, effects, compared, heavenwardive,
|[adult, one, along, book, journey, two, saved, heartache, refer, over, children, lot, exsoninlaw, child, inlaw]
+----------------------------------------------------------------------------------------------------------------------------------------------------------------------
only showing top 20 rows
```

Figure 1. processed data frame after cleaning

The FP-Growth algorithm, a widely used technique for mining frequent itemsets in large datasets, was applied with minimum support of 0.01 and a minimum confidence of 0.2. It is worth mentioning that minimum support refers to the proportion of transactions in the dataset that must contain an itemset for it to be considered frequent. It ensures that only itemsets that appear frequently are selected, and minimum confidence is the measure of the likelihood that a rule is accurate, given the presence of the antecedent. It indicates how often the rule has been found to be true in the data.

Presented below in figure 2 are the word cloud and bar plot visualizations, showcasing the most frequently mentioned words in reviews. It is observable that book and read are the most frequent words in reviews.

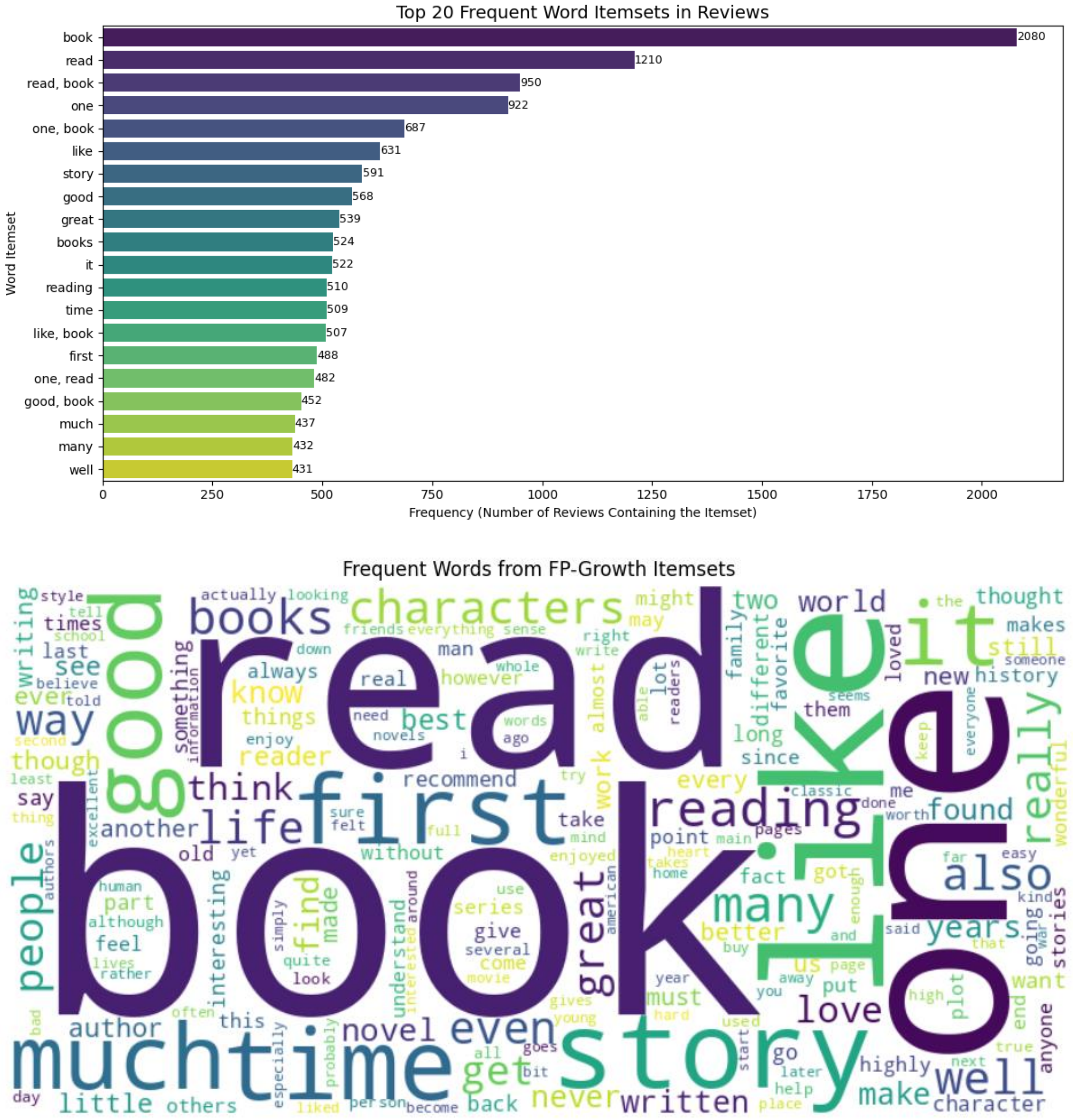
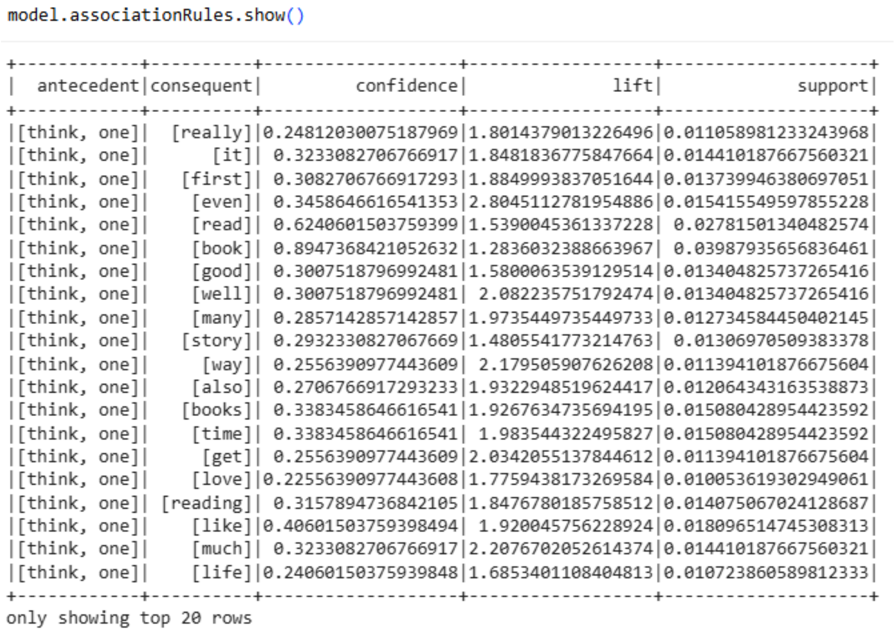


Figure 2. Frequent words in reviews bar plot and word cloud.

Association rules also derived from the dataset after model implementation. These rules provide meaningful co-occurrence patterns among common review words, reflecting readers' shared language when expressing opinions about books. Each rule consists of an antecedent, representing a combination of words that suggest the presence of another word, known as the consequent. As shown in Figure 3, for example, the rule indicating that reviews containing the words [think, one] ⇒ [book] has a confidence of 0.89 and a lift of 1.28, meaning that when the words "think" and "one" appear together, there is a strong likelihood that the word "book" also appears in the same review. Similarly, the rule [think, one] ⇒ [like] (confidence = 0.41, lift = 1.92) suggests that reviews containing "think" and "one" frequently also include “like”, reflecting a common sentiment pattern in how readers express their opinions.



```
model.associationRules.show()
```

```
+------------+----------+-------------------+------------------+--------------------+
|  antecedent|consequent|         confidence|              lift|             support|
+------------+----------+-------------------+------------------+--------------------+
|[think, one]|  [really]|0.24812030075187969|1.8014379013226496|0.011058981233243968|
|[think, one]|      [it]| 0.3233082706766917|1.8481836775847664|0.014410187667560321|
|[think, one]|   [first]| 0.3082706766917293|1.8849993837051644|0.013739946380697051|
|[think, one]|    [even]| 0.3458646616541353|2.8045112781954886|0.015415549597855228|
|[think, one]|    [read]| 0.6240601503759399|1.5390045361337228| 0.02781501340482574|
|[think, one]|    [book]| 0.8947368421052632|1.2836032388663967| 0.03987935656836461|
|[think, one]|    [good]| 0.3007518796992481|1.5800063539129514|0.013404825737265416|
|[think, one]|    [well]| 0.3007518796992481| 2.082235751792474|0.013404825737265416|
|[think, one]|    [many]| 0.2857142857142857|1.9735449735449733|0.012734584450402145|
|[think, one]|   [story]| 0.2932330827067669|1.4805541773214763| 0.01306970509383378|
|[think, one]|     [way]| 0.2556390977443609| 2.179505907626208|0.011394101876675604|
|[think, one]|    [also]| 0.2706766917293233|1.9322948519624417|0.012064343163538873|
|[think, one]|   [books]| 0.3383458646616541|1.9267634735694195|0.015080428954423592|
|[think, one]|    [time]| 0.3383458646616541| 1.983544322495827|0.015080428954423592|
|[think, one]|     [get]| 0.2556390977443609|2.0342055137844612|0.011394101876675604|
|[think, one]|    [love]|0.22556390977443608|1.7759438173269584|0.010053619302949061|
|[think, one]| [reading]| 0.3157894736842105|1.8476780185758512|0.014075067024128687|
|[think, one]|    [like]|0.40601503759398494| 1.920045756228924|0.018096514745308313|
|[think, one]|    [much]| 0.3233082706766917|2.2076702052614374|0.014410187667560321|
|[think, one]|    [life]|0.24060150375939848|1.6853401108404813|0.010723860589812333|
+------------+----------+-------------------+------------------+--------------------+
only showing top 20 rows
```

Figure 3. association rules

In the second step, a system finding frequent itemsets considering books as items is implemented. In this part the objective of the work is to understand the set of books reviewed by the same user. For this reason, the rows with null values in Title and id and review columns dropped out. The dataset then is grouped by User_id and considering set of book title. Afte that just those ids with more than 1 book are considered. FP-Growth algorithm is implemented with minimum support of 0.001 and a minimum confidence of 0.1. Presented below are top frequent books and pair books bar plot visualizations.

In figure 4 it is observable that pride and prejudice and also the hobbit has been the most reviewed books, and the same books reviewed by the same users multiple times.

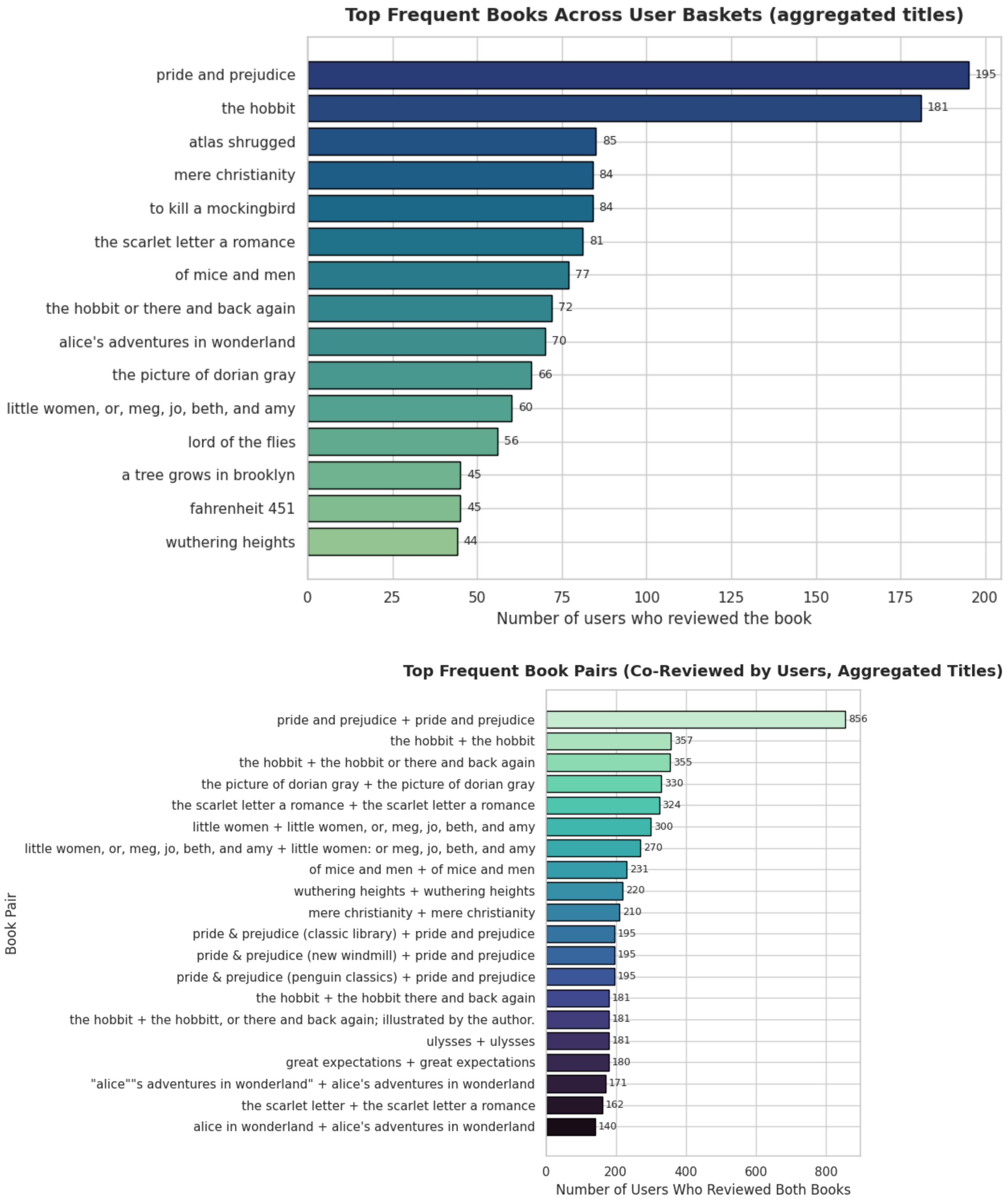
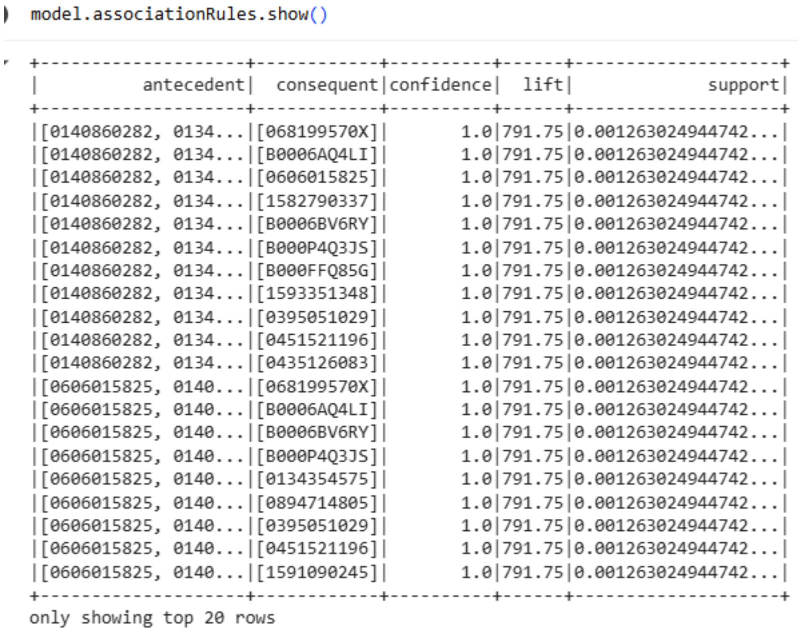


Figure 4. top frequent books and pair books reviewed bar charts

The association rules in figure 5 reveal that certain books tend to be reviewed together by the same users. The very high confidence and lift values suggest strong relationships between these books —

meaning that if a user reviews one of them, they’re very likely to have reviewed the others as well. This points to clear co-reading patterns among users.



```
model.associationRules.show()
```

```
+--------------------+------------+----------+------+--------------------+
|          antecedent|  consequent|confidence|  lift|             support|
+--------------------+------------+----------+------+--------------------+
|[0140860282, 0134...|[068199570X]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[B0006AQ4LI]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[0606015825]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[1582790337]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[B0006BV6RY]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[B000P4Q3JS]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[B000FFQ85G]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[1593351348]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[0395051029]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[0451521196]|       1.0|791.75|0.001263024944742...|
|[0140860282, 0134...|[0435126083]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[068199570X]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[B0006AQ4LI]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[B0006BV6RY]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[B000P4Q3JS]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[0134354575]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[0894714805]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[0395051029]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[0451521196]|       1.0|791.75|0.001263024944742...|
|[0606015825, 0140...|[1591090245]|       1.0|791.75|0.001263024944742...|
+--------------------+------------+----------+------+--------------------+
only showing top 20 rows
```

Figure 5. association rules

Conclusion

In this project FPGrowth algorithm implemented to analyze book review in two perspectives: frequent word combinations in reviews and the set of books reviewed by a same user. The analysis uncovered insightful connections between words and books, highlighting common reading habits and ways readers express their opinions. These results show that the FP-Growth algorithm is a powerful tool for finding valuable patterns in large datasets, helping to enhance book recommendations, organize titles more effectively, and better understand reader preferences.

References

Ahn, Kwang-Il, and Jae-Yearn Kim. "Efficient mining of frequent itemsets and a measure of interest for association rule mining." Journal of Information & Knowledge Management 3.03 (2004): 245-257.

Aldino, Ahmad Ari, et al. "Comparison of market basket analysis to determine consumer purchasing patterns using fp-growth and apriori algorithm." 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE). IEEE, 2021.

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submited by me or any other person for assessment on this or any other course of study.