PROJECT STAGE – 4

Team Members: Aravind Soundararajan, Krishnan Rajagopalan, Palaniappan Nagarajan

1. **How did you combine the two tables A and B to obtain E? Did you add any other table? When you did the combination, did you run into any issues?**

First, we use the matcher M (which in our case is the logistic regression model) to predict a match for the 2048 tuple pairs that survived the blocker stage. The matcher M returns 0 if the tuple pairs do not match and 1 if they match.

We generate the schema for Table E. Since we have a similar schema for both A and B, table E would have the common schema.

Attribute Name	Datatype	Description
Title	String	Title of the book (including the Edition number) as given in the source from which the book details are taken.
Authors	String	Names of all the persons listed under the author's section along with the authors. This includes the translator, the illustrator etc.
Genres	String	The category of books under which the book can be classified such as Horror, Comedy etc. A single book can be classified under multiple genres.
Publishing Date	String	Date of publishing.
Pages	Integer	Number of pages in the book.
Publisher	String	Name of the publisher(s) as listed in the source.
Language	String	Language.

We do not use any other tables apart from A and B for merging. In our case, table A is the CS-838-goodreads.csv and table B is the CS-838-bookdepository.csv file.

We consider tuples in the Book Depository table and tuples in the Goodreads table that are not part of the set of matching tuples and add it to the table E (dangling tuples). Next, we take all the tuple pairs that are a match. For every record in table A, collect all records in B that are identified as match by Magellan. Merge the data in the single record in table A and the collection of records in table B to result in a canonical record to be included in the table E. The merging is done as follows:

Attribute	Merging of Attribute	Logic Behind Merging
Book Title	Take the longest title.	The longest title would be the most descriptive of all the titles.
Author	The list having more number of authors. If lists have same number of values, take name from Book Depository.	One of the websites may specify all the authors of the book while the other specifies only the main author. However, we give preference to Book Depository website as it is a more reliable commercial source.
Genre	Union of values from both the tables.	We would like to list as many genres as possible for every book. So even if first table has genre 'Romance' and the second table has genre 'Romantic Comedy', we would like to list both.
Publishing Date, Pages, Publisher, Language	The values are taken from Book Depository table. If that values are missing, we take it from Goodreads table.	Book Depository is a commercial website and hence the data is expected be more reliable than Goodreads which crowdsources data from the users of the application.

Missing Values: When there are missing values in any of the two tuples to be merged, we chose the tuple with value being present. In the case where the data was missing from both the tuples no action was taken.

Issues: Initially, we decided to take the Book Title from among matching tuples giving preference to Book Depository table. However, we had multiple Book Depository tuples that matched because of which we got multiple records in the table E with the same book title. Hence, we considered the book title with the maximum length from among the matching tuples.

2. Statistics on Table E

Parameter	Value
Number of tuples in Table E	7098
Number of variables	8
Total Missing (%)	4.2%
Total size in memory	342.2 KiB
Average record size in memory	64.0 B

Authors distinct values	2037
Publisher distinct values	955

Schema of Table E:

Title	Authors	Genres	Publishing Date	Pages	Publisher	Language

Sample tuples from E *(at least 4)*:

Title	Authors	Genres	Publishing Date	Pages	Publisher	Language
Call Me By Your Name	Andre Aciman	Contemporary Fiction:Erotic Fiction:Romance:Romance Books:Adult & Contemporary Romance:Adult & Contemporary Romance	21 Mar 2018	256	ATLANTIC BOOKS	English
Big Little Lies	Liane Moriarty	Contemporary Fiction:Crime:Crime Fiction:Thriller Books:Thrillers:Romance:Romance Books:Adult & Contemporary Romance:Adult & Contemporary Romance:Family & Relationships	07 May 2015	496	Penguin Books Ltd	English
The Dressmaker	Rosalie Ham	Contemporary Fiction:Romance:Romance Books:Historical Fiction	01 Jun 2016	320	Profile Books Ltd	English
After You	Jojo Moyes	Contemporary Fiction:Adult & Contemporary Romance:Adult & Contemporary Romance	15 Sep 2016	448	Penguin Books Ltd	English

Code for Merge:

```
def merge_titles(records):
    length = 0
    for record in records:
        cur_title = record['Title'].strip()
        if len(cur_title) > length:
            length = len(cur_title)
            title = cur_title
    return title
```

```
def merge_authors(records):
    authors = list()
    length = 0
    for record in records:
        current_authors = record['Authors'].strip().split(":")
        if len(current_authors) > length:
            length = len(current_authors)
            authors = ":".join(current_authors)
    return authors

def merge_genres(records):
    genres = set(records[-1]['Genres'].strip().split(":"))
    for i in range(len(records) - 1):
        cur_genres = records[i]['Genres'].strip().split(":")
        genres.union(cur_genres)

    return ":".join(list(genres))

def merge_publishing_dates(records):
    for record in records:
        pub_date = record['Publishing Date'].strip()
        if pub_date is not None and len(pub_date) > 0:
            return pub_date

def merge_pages(records):
    for record in records:
        pages = record['Pages'].strip()
        if pages is not None and len(pages) > 0:
            return pages

def merge_publishers(records):
    for record in records:
        publishers = record['Publisher'].strip()
        if publishers is not None and len(publishers) > 0:
            return publishers

def merge_languages(records):
    for record in records:
        languages = record['Language'].strip()
        if languages is not None and len(languages) > 0:
```

```
        return languages

    def merge_tuples(records):
      final_title = merge_titles(records)
      authors = merge_authors(records)
      genres = merge_genres(records)
      publishing_date = merge_publishing_dates(records)
      pages = merge_pages(records)
      publisher = merge_publishers(records)
      language = merge_languages(records)
          final_record = (final_title, authors, genres, publishing_date, pages, publisher, language)

      return final_record
```

What was the data analysis task that you wanted to do?

We have decided to perform OLAP operations on the table. We have performed Roll up, Slicing and Dicing operations on the data cube.

- Roll up: We find the count of the pages written by various authors under each genre type. Specifically, we find the sum of the number of pages of books listed under the following categories: 'Historical Romance', 'Contemporary Fiction', 'Thriller', 'Adult' and 'Drama'. So, we basically performed a roll-up operation on the 'Genres' dimension which had numerous categories like 'Adventure', 'Romantic Comedy', 'Horror' etc. that could be grouped into one of the major categories listed above.

- Slicing: We find the number of pages written by each author in the year 2016. We also get the number of pages written by E.L. James (the author of the Fifty Shades series) in all the years.

- Dicing: We find the number of books categorized under the genre 'Fiction' that have been written in the year 2016.

Accuracy Numbers:

- Roll up: We rolled up the OLAP cube on the 'Genres' dimension. We get the following table after the roll-up:

Genre_Type	Title	Authors	Genres	Publishing Date	Pages	Publisher	Language
	169	169	0	154	169	121	123
Adult	1868	1867	1868	1845	1868	1627	1246
Contemporary Fiction	1256	1256	1256	1256	1256	1256	1256
Drama	10	10	10	9	10	9	9
Historical Romance	1526	1526	1526	1513	1526	1513	1469
Others	1869	1868	1869	1810	1869	1750	1456
Thriller	149	149	149	145	149	143	139

- Slicing: We find the number of pages written by each author in the year 2016.

	Authors	Average pages
0	A N Roquelaure	434
1	A. Meredith Walters	321
2	A. Zavarelli	314
3	Abbi Glines	592
4	Abby Clements	368
5	Alessandra Torre	413
6	Alexa Riley	143
7	Alexandra Bracken	496
8	Alice Clayton	336
9	Alison G. Bailey	479
10	Alora Kate	306
11	Alora Kate;Silvia Curry	23
12	Aly Martinez	340
13	Amanda Quick	423
14	Amber A. Bardan	352
15	Amelia Hutchins	409
16	Amy Leigh Simpson	418

Similarly, we generate the table for the number of pages written by E.L. James for each year:

	year	Average pages
0	2011	7284
1	2012	2895
2	2015	576

- Dicing: We find the number of books categorized under the genre 'Fiction' that have been written in the year 2016 and found that 194 books categorized under some form of Fiction such as 'Contemporary Fiction' were written that year.

What did you learn/conclude from your data analysis? Were there any problems with the analysis process and with the data?

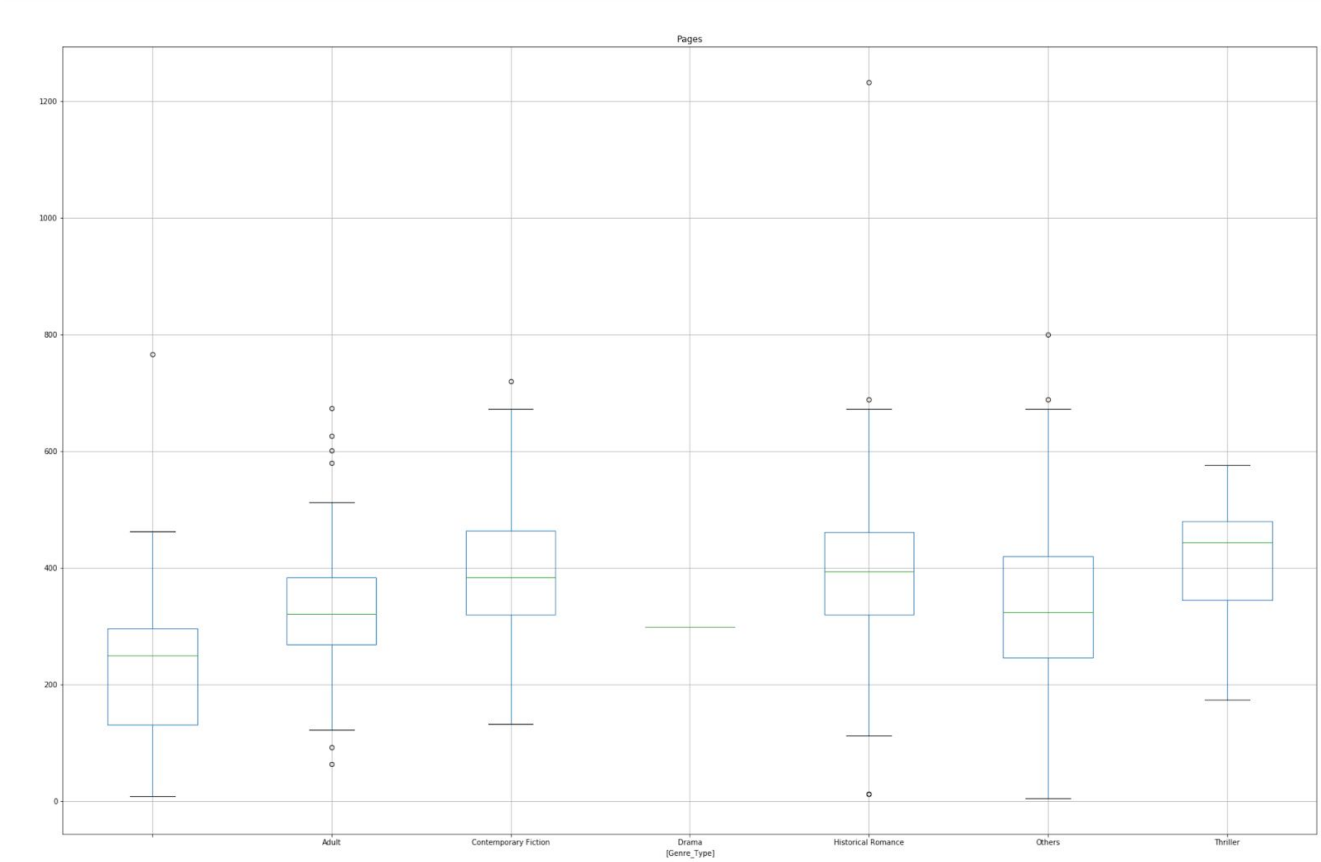
Each of the above operations gave us a useful insight into the data:

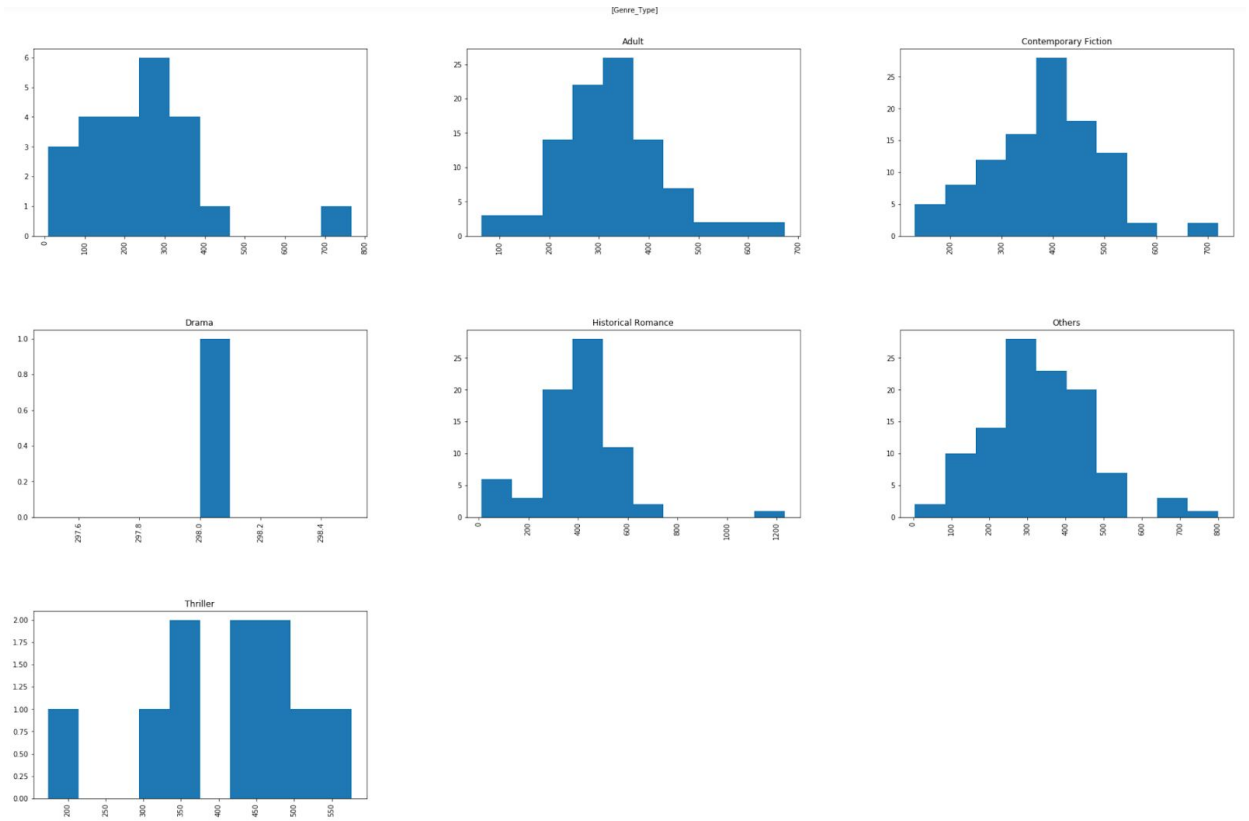
- Roll up: We found that the most number of books that were written are categorized under the genre 'Adult'. In order to study more about which genre of books were published in which year, we generated some plots:

	Pages		
	min	mean	max
Genre_Type			
	9	246.086957	766
Adult	64	328.073684	673
Contemporary Fiction	132	381.653846	720
Drama	298	298.000000	298
Historical Romance	12	396.718310	1232
Others	5	328.731481	800
Thriller	174	410.300000	576

This is the statistics for the number of pages that were written under each of the main genre categories in the year 2016. For this year, we see that on an average, the number of pages associated with the books under 'Thriller' is the maximum. However, the book with the maximum number of pages in the category of 'Historical Romance' is far greater than the minimum number of pages. This tells us that there may be huge variations in the styles of the authors of the 'Historical Romance' group.

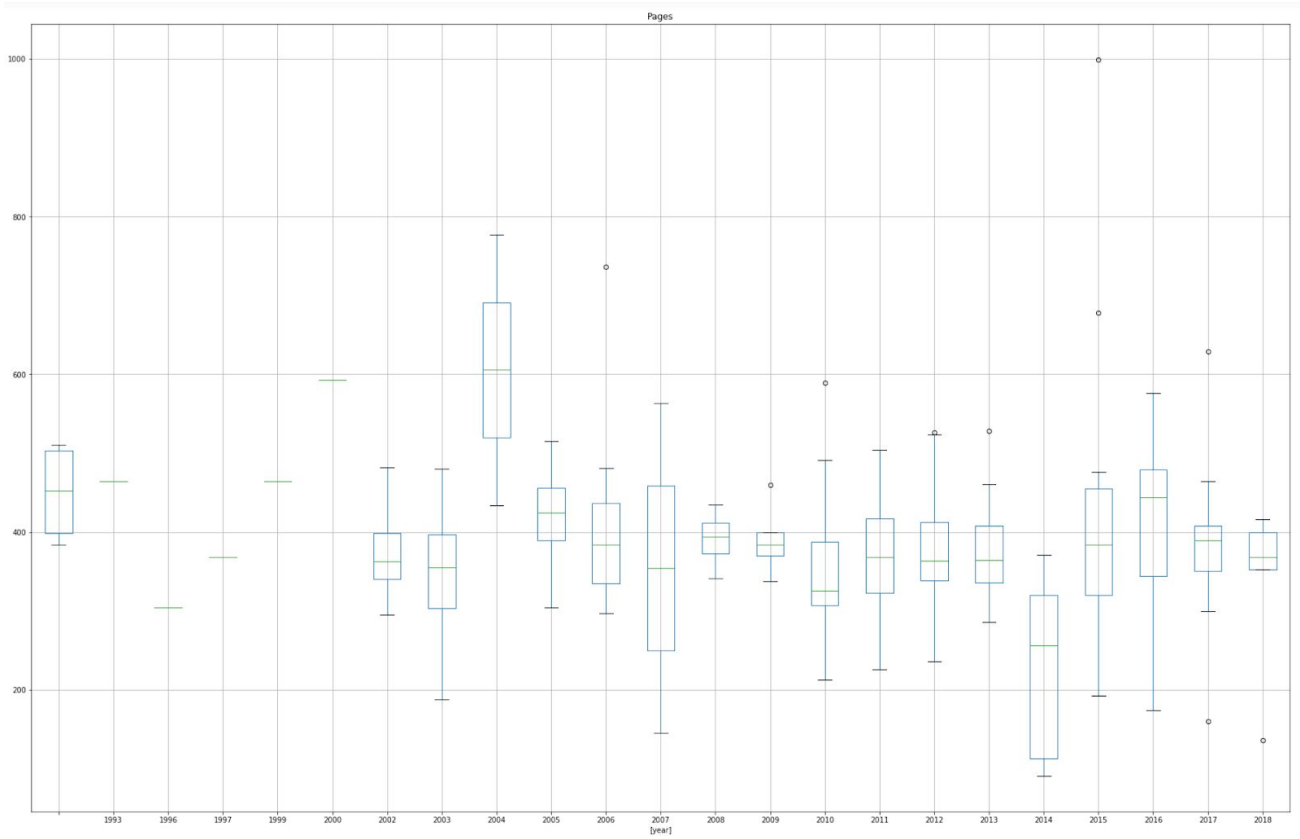
It can visualized graphically as follows:

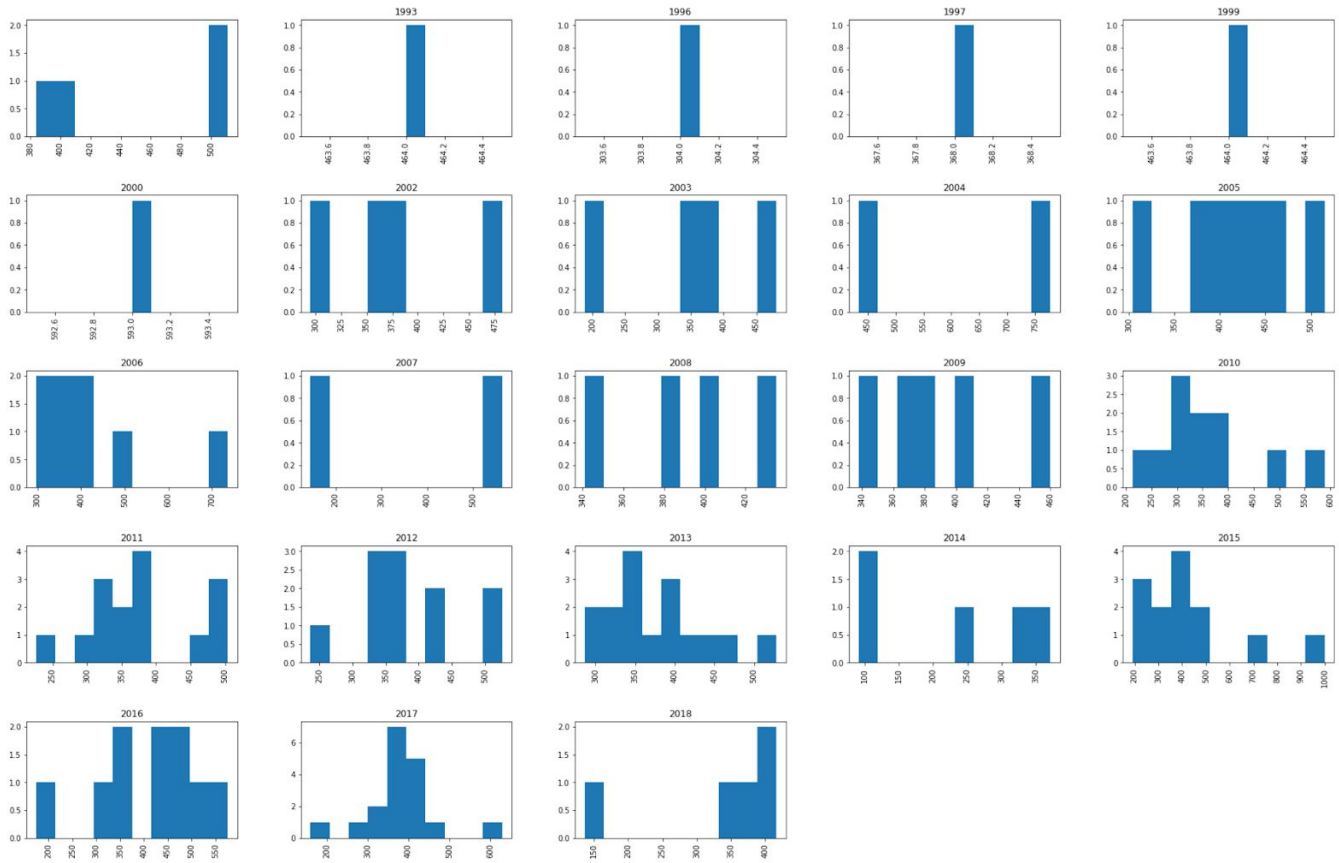




We can see that there is a very marked decrease in the number of pages of ‘Historical Romance’ published through the year. Also, for only a very short period of time (around mid of the year), there were a large number of books related to the ‘Drama’ genre that were published. It might have been a craze at that time and the trend might have faded out quickly.

We also performed an analysis on the number of pages of ‘Thriller’ books that have been published over the years. We see that ‘Thriller’ books were most popular around 2004 and least popular around 2014 (exactly 10 years later).


Pages
1000
800
600
400
200
1993
1996
1997
1999
2000
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
[year]



- Slice: While performing the slice operation considering the number of pages published by E.L. James, we see that she published the maximum initially at the start of the series which put her at the peak of her career with the first book of the Fifty Shades series. In recent years, she has not written extensively.

- Dice: We see that 194 books were published under the genre of 'Contemporary Fiction' in 2016. However, the number of pages published under the very same genre is very less. This shows that the books published during this time under this genre had very less number of pages and authors did not choose to write extensively.

Problems with analysis:

- Missing values introduced problems in deriving insights. Also, they created problems while processing with dataframe as some of the default values had data type issues.
    - Aggregating to sum an integer column with 'Nan' values resulted in concatenation.

    - Finding mean value of an integer column resulted in accounting ‘Nan’ values as valid values thereby largely reducing the average (mean) value.
    - ‘Nan’ values in string columns were interpreted as float values and generated exceptions.
- ‘Genres’ column stores a list of values. Since there is no efficient way to identify duplicates (except naive string matching), many similar genres were present for each record (eg. Romance, Romantic Drama, Contemporary Romance etc)
- ‘Authors’ column also contains multiple names, but since the convention across the two sources is not consistent it created conflicts while merging data in the ‘Authors’ column.

If you have more time, what would you propose you can do next?

We would have matched the book records with data from the Book Awards database and would have used a classification algorithm for predicting if a newly published book would win any awards or not.

We would have also included the user rating given to each book and would have used a suitable machine learning algorithm to predict the rating that would be given by a customer for any book and recommend a new book for him given the genres he has already rated highly for.

We have performed all the operations in memory and would like to extend our use case to Map-Reduce in Hadoop clusters.