Final Project Proposal

Team name: Kakao

Team composition (names only):

- Dongyeon Kang
- Jay Patel
- Jiho Lee
- Jongmin Chung
- Junho Lee
- Seongeun Kim
- Sunwoo Kim
- Yong Jun Choi

CREDIT Statement:
https://www.elsevier.com/researcher/author/policies-and-guidelines/credit-author-statement
Each roll will be evenly distributed among teammates based on the following criteria below:

- Conceptualization
- Methodology
- Software
- Validation
- Formal Analysis
- Investigation
- Resources
- Data Curation
- Writing- Original Draft / Review/ Editing
- Visualization
- Supervision
- Project Administration

Abstract:

The objective of this study is to develop a genre-specific book recommendation system by analyzing the Amazon book review dataset. Utilizing techniques learned in class, such as Alternating Least Squares (ALS), the system aims to personalize recommendations for users based on their preferences and reading history. Through meticulous data analysis and algorithm implementation, this research seeks to enhance user experience by providing tailored book suggestions across various genres, thereby improving discovery and engagement on the platform. The study will explore the dataset's characteristics, implement the ALS algorithm, and evaluate the system's effectiveness in delivering accurate and relevant recommendations.

Motivation:

Our team shares a strong passion for reading and literature. Drawing inspiration from our coursework in data analytics and machine learning, we have set our sights on the challenge of crafting a book recommendation system tailored to the vast array of genres that literature has to offer. We aim to apply the Alternating Least Squares (ALS) technique and other learned methodologies to uncover personalized book suggestions from the intricate web of reader preferences and histories. This project is not just making a platform better; it is about making a reading experience better for everyone. We aim to ensure that every new book a reader discovers feels as engaging as their past favorites, helping them feel a closer bond with the world of books.

Problem statement:

Understanding and predicting trends in popular books and individual reading preference draw a challenge that requires further research. To explore this idea effectively, we intend to adopt two focused strategies:

- Book recommendations:
    - By preprocessing the data of the reviews submitted by individual Amazon Books users, the objective is to enhance the book recommendations for each user. By analyzing ratings and feedback provided by users, the goal aims to identify patterns and preferences personalized for individual users.
- Trend Analysis:
    - Analyzes the changing popularity of book genres over time using the Amazon books reviews dataset. It looks at review amounts, ratings, and sentiments to predict future popular genres, helping publishers and authors decide which books to focus on.

Description of raw dataset:

- Books_ratings
    - ID : id of the book
    - Title
    - Price
    - User_id : id of user who rate the book
    - Profile Name : name of user who rate the book
    - Review/helpfulness: helpfulness rating of the review
    - Review/score: rating score from 0 to 5
    - Review/time: time of given the review
    - Review/summary: the summary of text review
    - Review/text: the full text of a review

- Book_data
    - Title
    - Description: description of book
    - Authors: name of book authors
    - Image: url for book cover
    - Preview Link: link to access this book on google books
    - Publisher: name of publisher
    - Published Date
    - Info link: link to get more information on google books
    - Categories: genre of the book
    - Ratings Count: average rating for book

Link to datasets:

- https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews

Methods utilized:

- RDD/Dataframe Transformation
    - Filter and Sort the ratings by genre
    - Search keywords in ratings for book recommendation

- ~~Hyperparameter Tuning~~

    ~~Genre Classification and rating prediction.~~

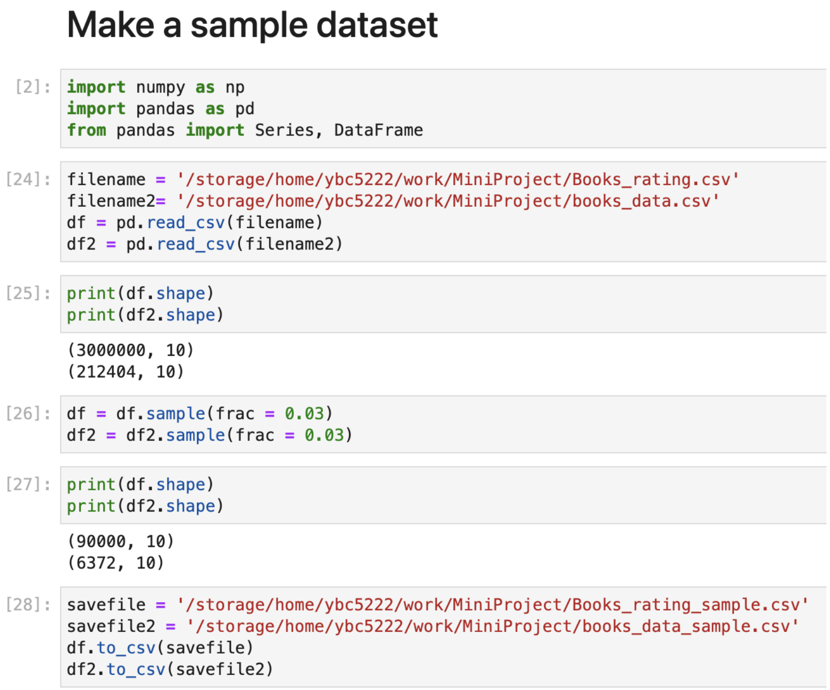
    ~~Do CNN or Neural Network~~

- Term Frequency-Inverse Document Frequency (TF-IDF)
    - Calculate the importance of a word in the dataset relative to the other words by measuring the similarity between the content and user preferences
    - Based on the calculated TF-IDF scores for each user, we can sort the books with specific user profile and recommend the top N books to the user
    - The performance of this model will be evaluated using appropriate evaluation metrics.
- Alternating Least Squares (ALS)
    - Matrix factorization: convert dataset into a user-item interaction matrix
    - Update the user and item matrices using the ALS optimization algorithm
    - Repeat the process until convergence of iterations is reached
    - Performance will be evaluated using metrics like Mean Absolute Error or Root Mean Squared Error.

Why do we require ICDS?

The use of the ICDS is crucial for this project due to the substantial size of the Amazon book review dataset (2.86GB). This dataset encompasses a vast collection of reviews written by numerous users over an extensive period, ranging from May 1996 to July 2014. Preprocessing and analyzing such a large-scale dataset, as well as developing a recommendation system, would be computationally intensive and challenging to accomplish on a local machine with limited resources. Leveraging the high-performance computing capabilities of the ICDS enables efficient data processing, model training, and deployment of the recommendation system, thereby overcoming the limitations of traditional computing environments.

Weekly Update Summary:



Make a sample dataset

```
[2]:  import numpy as np
      import pandas as pd
      from pandas import Series, DataFrame
```

```
[24]: filename = '/storage/home/ybc5222/work/MiniProject/Books_rating.csv'
      filename2= '/storage/home/ybc5222/work/MiniProject/books_data.csv'
      df = pd.read_csv(filename)
      df2 = pd.read_csv(filename2)
```

```
[25]: print(df.shape)
      print(df2.shape)

      (3000000, 10)
      (212404, 10)
```

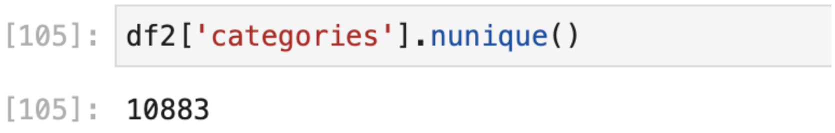
```
[26]: df = df.sample(frac = 0.03)
      df2 = df2.sample(frac = 0.03)
```

```
[27]: print(df.shape)
      print(df2.shape)

      (90000, 10)
      (6372, 10)
```

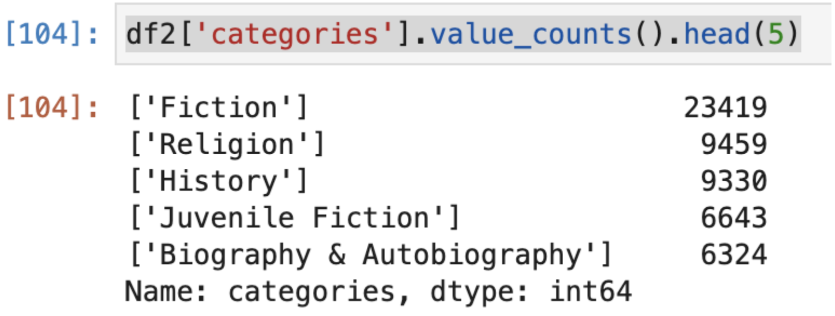
```
[28]: savefile = '/storage/home/ybc5222/work/MiniProject/Books_rating_sample.csv'
      savefile2 = '/storage/home/ybc5222/work/MiniProject/books_data_sample.csv'
      df.to_csv(savefile)
      df2.to_csv(savefile2)
```

We made a random sample dataset consisting of 3 percent of the full dataset, to develop the code faster with a smaller dataset to make sure that the code works and we got the right concept. We will use the sample dataset for our weekly update report and we will be using the full dataset for our final report submission after we validate all of our codes.



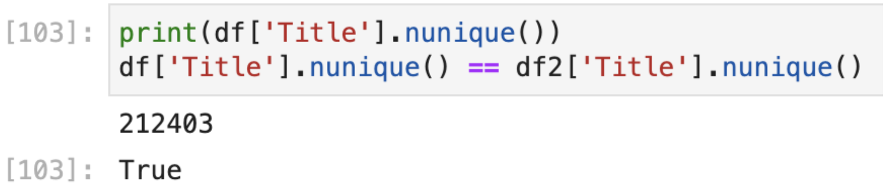
```
[105]: df2['categories'].nunique()

[105]: 10883
```



```
[104]: df2['categories'].value_counts().head(5)

[104]: ['Fiction']                          23419
       ['Religion']                          9459
       ['History']                           9330
       ['Juvenile Fiction']                  6643
       ['Biography & Autobiography']         6324
       Name: categories, dtype: int64
```

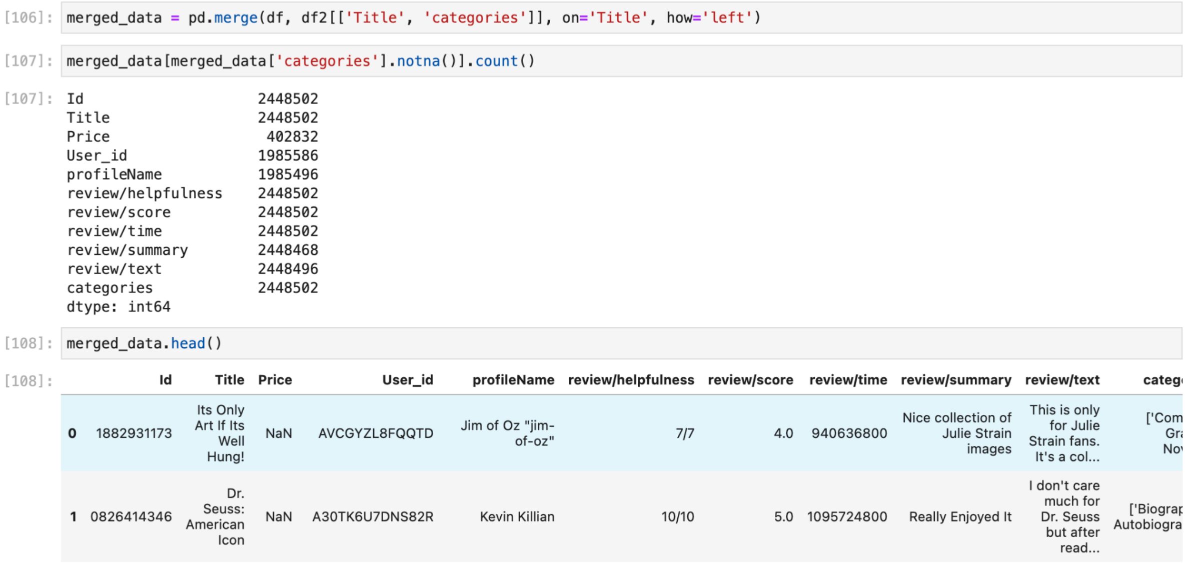


```
[103]: print(df['Title'].nunique())
       df['Title'].nunique() == df2['Title'].nunique()

       212403
[103]: True
```

We’ve checked that ‘Fiction’ is the most popular category among the books.

We’ve also checked that the number of unique values in the two dataset is the same. Using this fact, we have merged the two dataset using the ‘Title’ column.



```
[106]: merged_data = pd.merge(df, df2[['Title', 'categories']], on='Title', how='left')
```

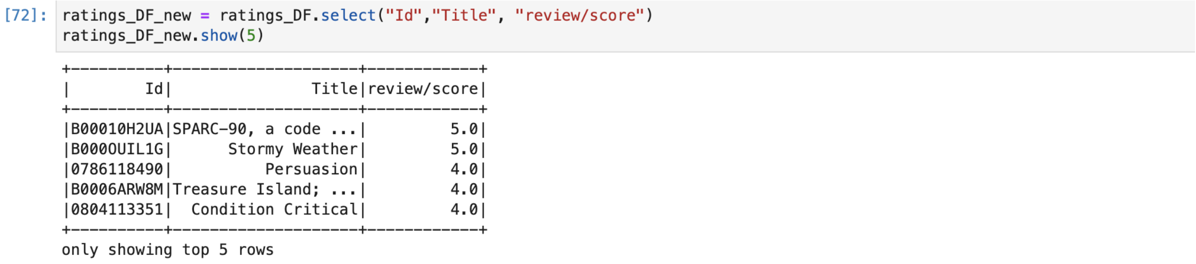
```
[107]: merged_data[merged_data['categories'].notna()].count()
```

```
[107]: Id                    2448502
       Title                 2448502
       Price                  402832
       User_id               1985586
       profileName           1985496
       review/helpfulness    2448502
       review/score          2448502
       review/time           2448502
       review/summary        2448468
       review/text           2448496
       categories            2448502
       dtype: int64
```

```
[108]: merged_data.head()
```

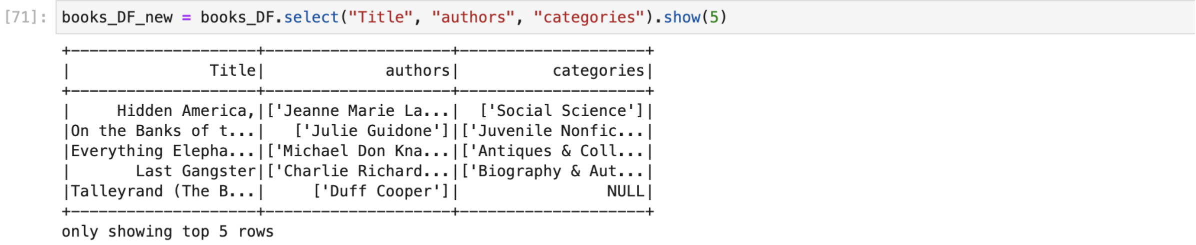
[108]:

	Id	Title	Price	User_id	profileName	review/helpfulness	review/score	review/time	review/summary	review/text	categ
0	1882931173	Its Only Art If Its Well Hung!	NaN	AVCGYZL8FQQTD	Jim of Oz "jim-of-oz"	7/7	4.0	940636800	Nice collection of Julie Strain images	This is only for Julie Strain fans. It's a col...	['Com Gra No
1	0826414346	Dr. Seuss: American Icon	NaN	A30TK6U7DNS82R	Kevin Killian	10/10	5.0	1095724800	Really Enjoyed It	I don't care much for Dr. Seuss but after read...	['Biograp Autobiogra



```
[72]: ratings_DF_new = ratings_DF.select("Id","Title", "review/score")
      ratings_DF_new.show(5)
```

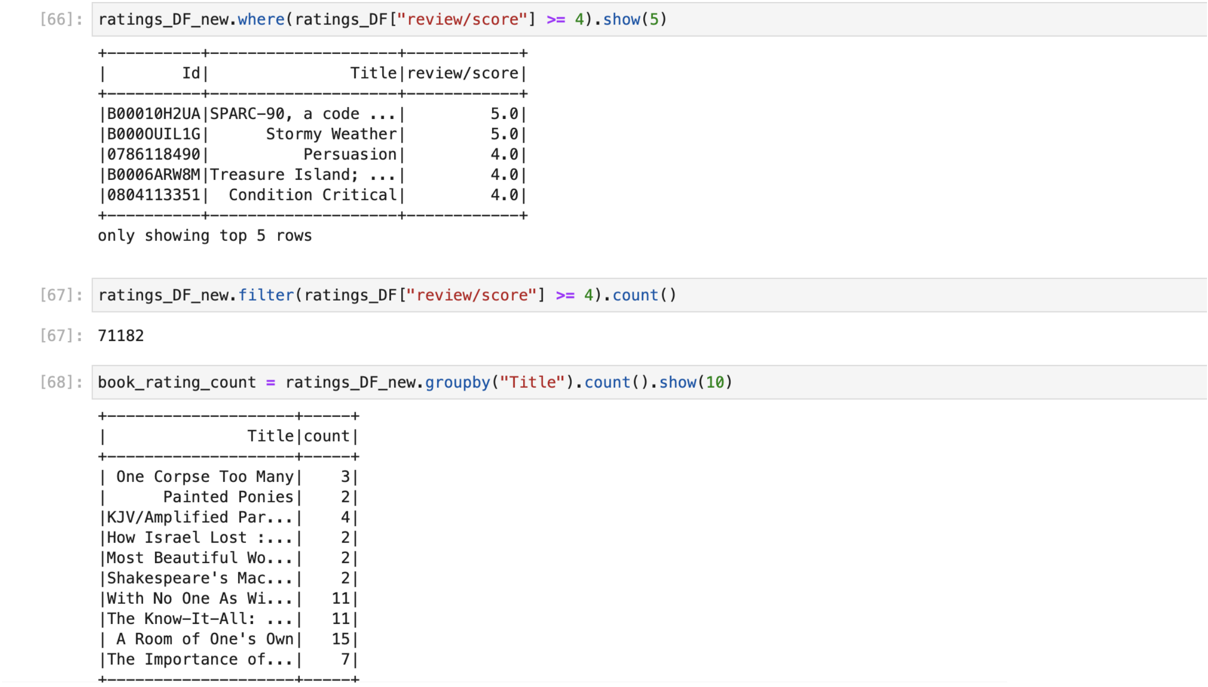
```
+----------+--------------------+------------+
|        Id|               Title|review/score|
+----------+--------------------+------------+
|B00010H2UA|SPARC-90, a code ...|         5.0|
|B000OUIL1G|      Stormy Weather|         5.0|
|0786118490|          Persuasion|         4.0|
|B0006ARW8M|Treasure Island; ...|         4.0|
|0804113351|  Condition Critical|         4.0|
+----------+--------------------+------------+
only showing top 5 rows
```



```
[71]: books_DF_new = books_DF.select("Title", "authors", "categories").show(5)
```

```
+--------------------+--------------------+--------------------+
|               Title|             authors|          categories|
+--------------------+--------------------+--------------------+
|     Hidden America,|['Jeanne Marie La...|  ['Social Science']|
|On the Banks of t...|   ['Julie Guidone']|['Juvenile Nonfic...|
|Everything Elepha...|['Michael Don Kna...|['Antiques & Coll...|
|       Last Gangster|['Charlie Richard...|['Biography & Aut...|
|Talleyrand (The B...|     ['Duff Cooper']|                NULL|
+--------------------+--------------------+--------------------+
only showing top 5 rows
```

We took the sample dataset and filtered out any columns that are not useful. In the ratings dataset we kept the ID of the book, the title and the review/score it received by readers. In the data sample we kept the title, author and categories/genre of the books.



```
[66]: ratings_DF_new.where(ratings_DF["review/score"] >= 4).show(5)

+----------+--------------------+------------+
|        Id|               Title|review/score|
+----------+--------------------+------------+
|B00010H2UA|SPARC-90, a code ...|         5.0|
|B000OUIL1G|      Stormy Weather|         5.0|
|0786118490|          Persuasion|         4.0|
|B0006ARW8M|Treasure Island; ...|         4.0|
|0804113351|  Condition Critical|         4.0|
+----------+--------------------+------------+
only showing top 5 rows
```

```
[67]: ratings_DF_new.filter(ratings_DF["review/score"] >= 4).count()

[67]: 71182
```

```
[68]: book_rating_count = ratings_DF_new.groupby("Title").count().show(10)

+--------------------+-----+
|               Title|count|
+--------------------+-----+
| One Corpse Too Many|    3|
|      Painted Ponies|    2|
|KJV/Amplified Par...|    4|
|How Israel Lost :...|    2|
|Most Beautiful Wo...|    2|
|Shakespeare's Mac...|    2|
|With No One As Wi...|   11|
|The Know-It-All: ...|   11|
| A Room of One's Own|   15|
|The Importance of...|    7|
+--------------------+-----+
```

With the filtered data we took the count of all the books that received a rating of 4 or higher. Also, took the count of all the ratings each book had received.