# Capstone Project - 2

# Team Space: Book Recommender System


**<u>Team Members:</u>**

**Saubhagya Verma**

**Harsh Mudgil**

**Tawheed Yousuf**

**Harshal Pawar**

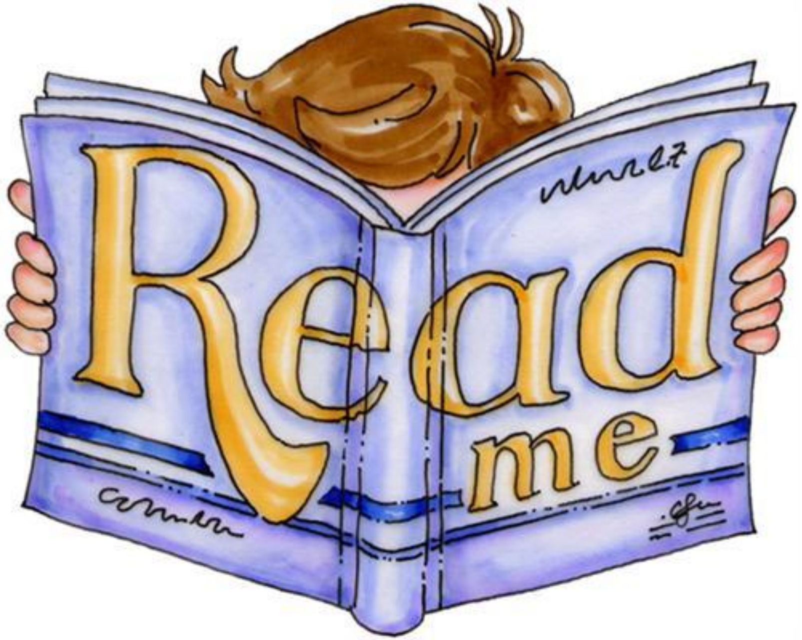
**Jimmi Kumar**

**Sai Krishna Reddy Palle**

*“Even the smallest seed of an idea can grow. It can grow to define or destroy you” - Cobb*

1. Problem Statement
2. Processing &

   Feature Engineering
3. Exploratory Data Analysis
4. Preparing Data For Models
5. Applying Models



# ➢ Problem Statement

- During the last few decades, recommender systems have taken more and more place in our lives. From e-commerce to online advertisement, recommender systems are today unavoidable in our daily online journeys.

- Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors.

- Recommender systems are algorithms aimed at suggesting relevant items to users. The main objective is to create a book recommendation system for users.

# Data Pipeline

- <u>Primary Inspection:</u> Observed irregularities in the data set and unique values for different columns

- <u>Processing & Feature Engineering:</u> Handled missing values, capped outliers and engineered features for further analysis. Data set was split for building different explicit rank based and implicit rank based recommender systems.

- <u>EDA:</u> Exploratory analysis was performed on columns like Book-Rating, Location, Book-Author to review trends and patterns emerging in the data set.

- <u>Applying Simple Models:</u> Models, based on mean ratings and K-Nearest-Neighbourhood Algorithm, were built to provide simple recommendations

# ➢ Data Pipeline

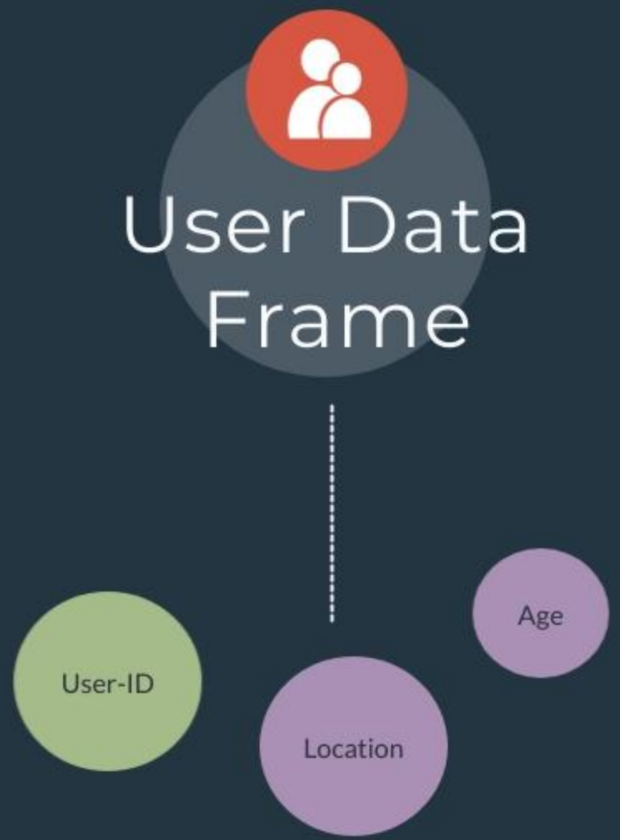
- **Applying Collaborative Filtering Model: SVD model based collaborative filtering system was built to provide recommendations based on user-user similarity, for explicitly ranked items.**

- **Applying Memory based Filtering: K-Nearest-Neighbourhood Algorithm, was used to make recommendations based upon user age, for implicitly rated items**

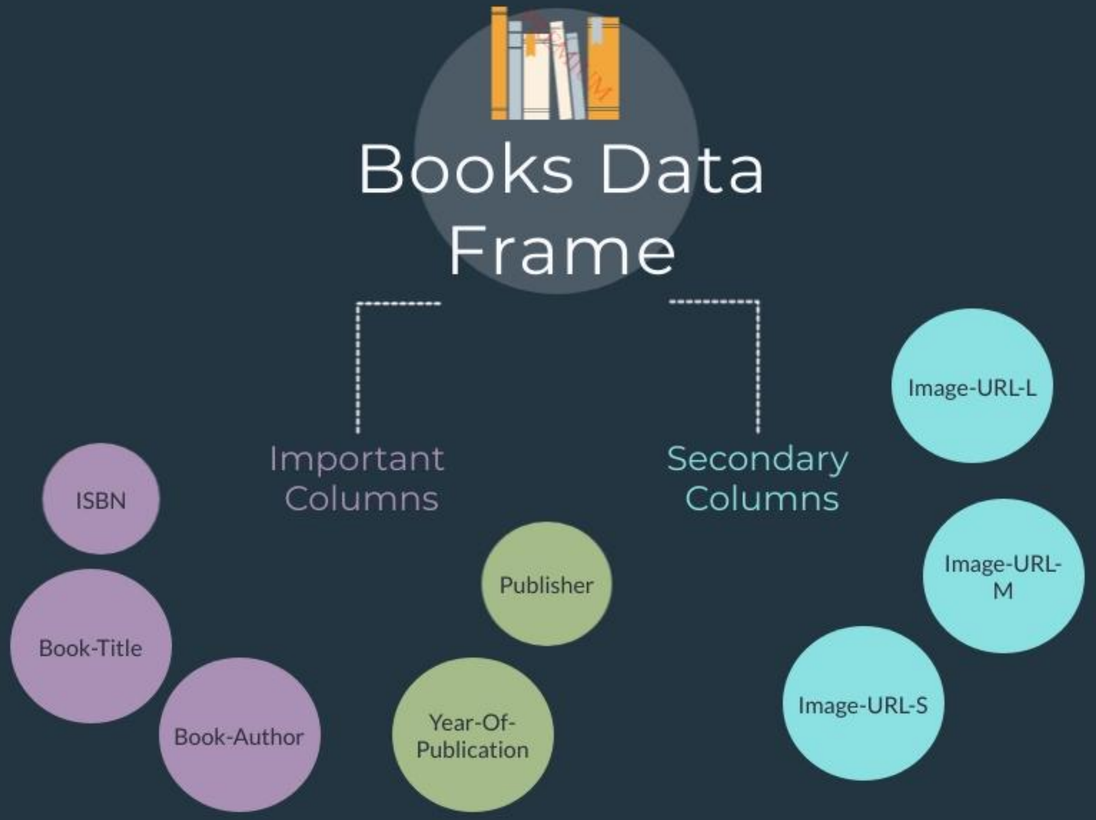
- **Content Based Solution: A model was built to recommend new books, based upon the content description of a user's past purchase.**

# Data Summary

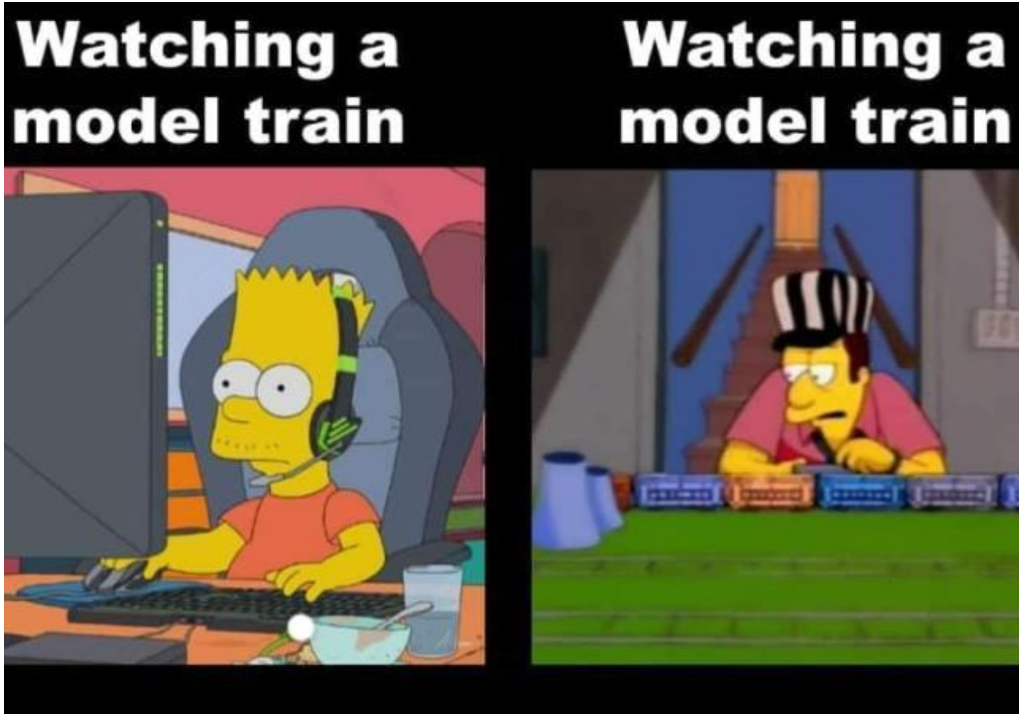
# Data Summary



# Data Summary



# Processing & Feature Engineering



# ➢ Processing & Feature Engineering

- **Feature Engineering on Location:** To analyse user country information, a function `get_country()` was developed to extract country information from Location column.

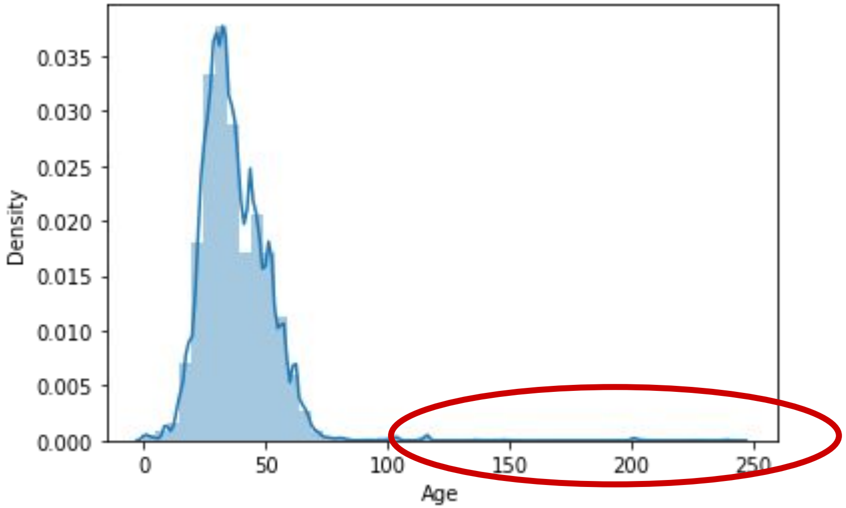
`get_country()`

- **Engineering Book Descriptions:** Description for Book-Titles, was fetched from Google Books API, in order to perform implicit, content similarity based recommendations

- **Feature Engineering Age:** Age Column was converted into bin, to better reflect preference of a user with respect to their age

# ➢ Processing & Feature Engineering

- **Capping Outliers:** User Age had outliers, which were capped randomly with values 90 and 100, in order to maintain the original distribution.



# ➢ Processing & Feature Engineering

- **<u>Handling Missing Values:</u>** `show_missing(y)` **was written to print missing value report for all columns of each data frame. A missing value for Publisher and Book-Author columns, was imputed with 'unknown'. Age column, with large number of missing values, was imputed with random numbers generated in the range of Median Absolute Deviation**

```
Missing Data Count
age_bins        247826
Age             245274
Image-URL-L          4
Publisher            2
dtype: int64
--------------------------
Missing Data Percentage
age_bins        27.03
Age             26.75
Image-URL-L      0.00
Publisher        0.00
dtype: float64
```

# ➢ Processing & Feature Engineering

- **Removing Duplicates:** Books of same title, had been authored by different authors. Author name and Book-Title were merged and finally duplicates were removed

- **IMDB Weighted Ratings:** Ratings were weighted, based on formula used by popular film ratings' website, IMDB. These ratings were, later, used for building simple models

$$W = \frac{Rv + Cm}{v + m}$$

where:

$W$ = Weighted Rating

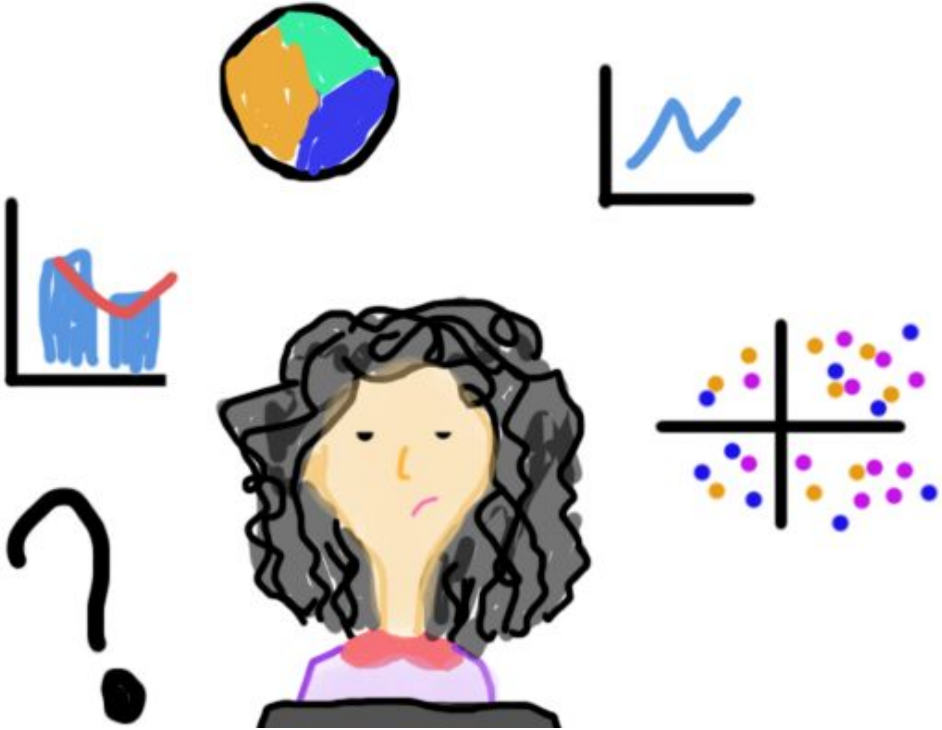
$R$ = average for the movie as a number from 0 to 10 (mean) = (Rating)

$v$ = number of votes for the movie = (votes)

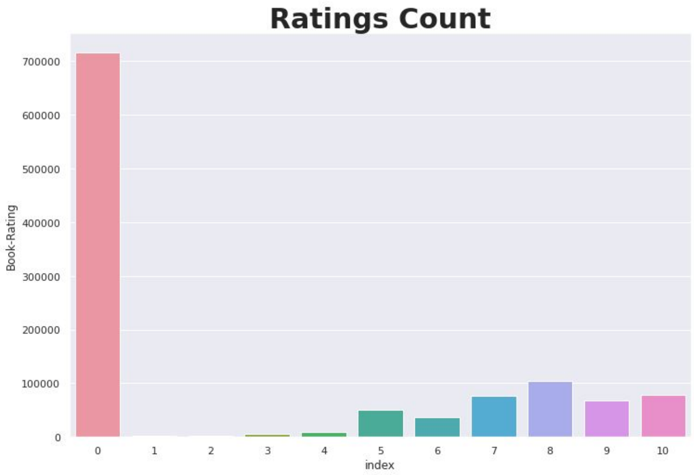
$m$ = minimum votes required to be listed in the Top 250 (currently 3000)

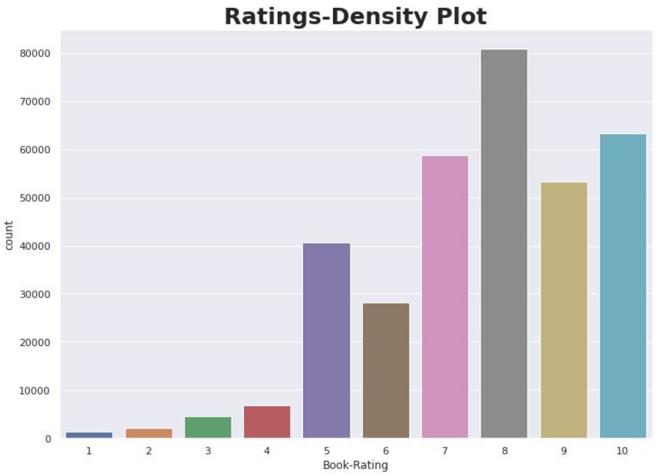
$C$ = the mean vote across the whole report (currently 6.9)

# Exploratory Data Analysis



# **Primary EDA:** Ratings, (Explicit + Implicit) vs Explicit

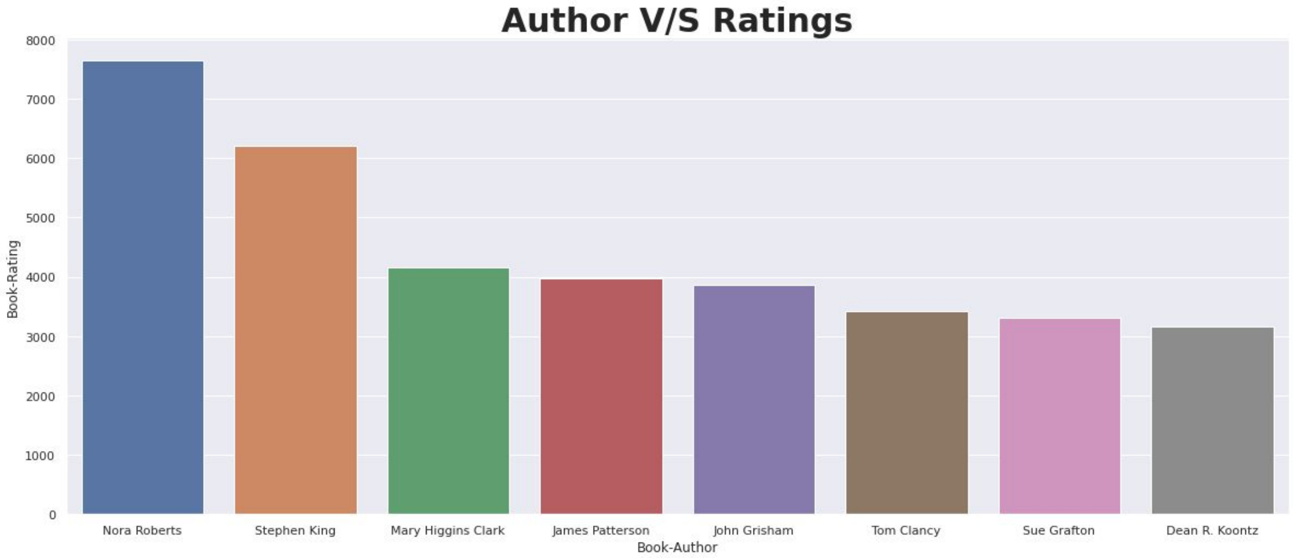



## ➢ Insights

- **We can see, by combining the implicit ratings with, explicit one's, the distribution of ratings becomes heavily skewed**

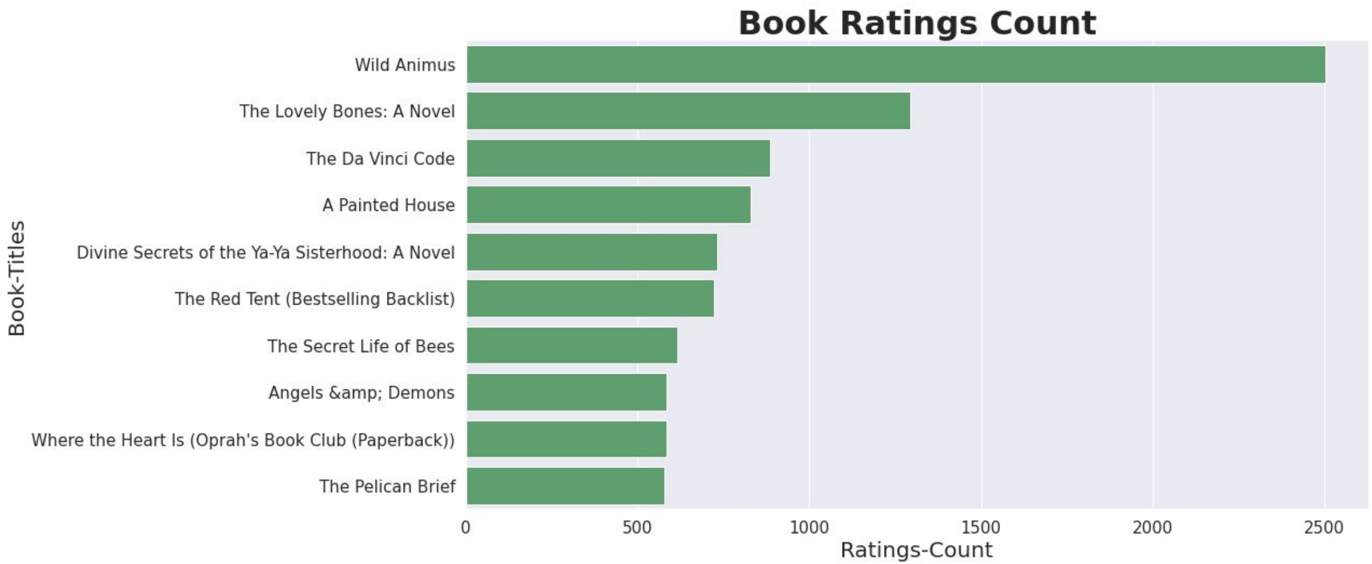
# Primary EDA: Author vs Ratings




## ➢ Insights

- Here, we can observe, most frequently rated Authors.
- Most frequently rated author is Nora Roberts, followed by Stephen King

# Primary EDA: Most Frequently Rated Books




## ➢ Insights

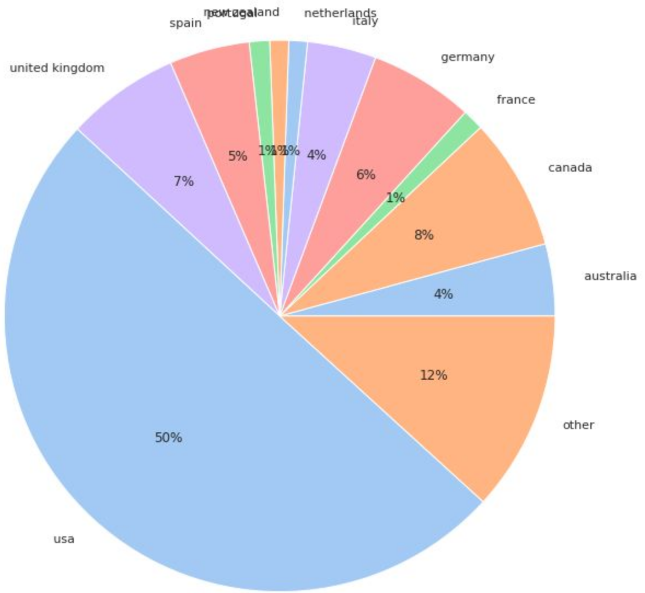
- Here, we are able to observe, most frequently rated books by the users.
- Most frequently rated book, happens to be Wild Animus

# **Primary EDA:** Country Representation in the Dataset

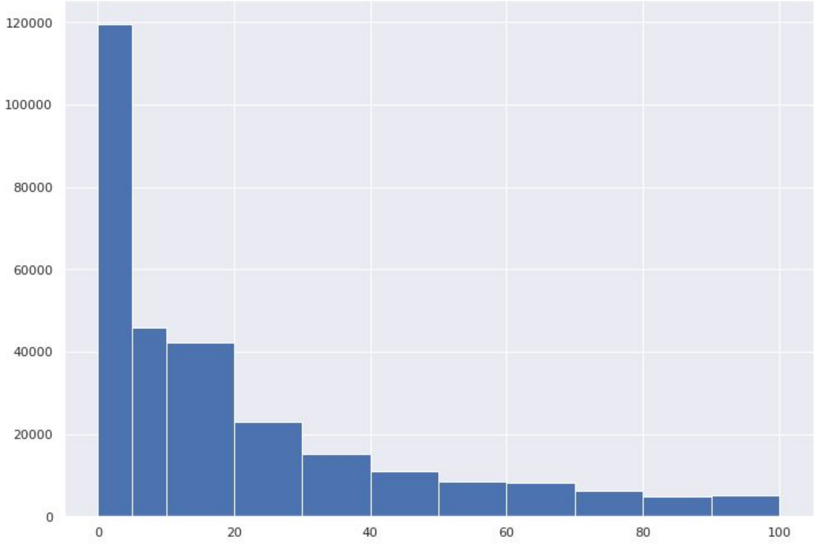
## ➢ Insights

- Most customers are from the United States of America, followed by Canada, United Kingdom and Germany
- *Countries with less than 1% customers are labelled as other

Country Representation in the Data Set



# Primary EDA: Age vs Rating Density



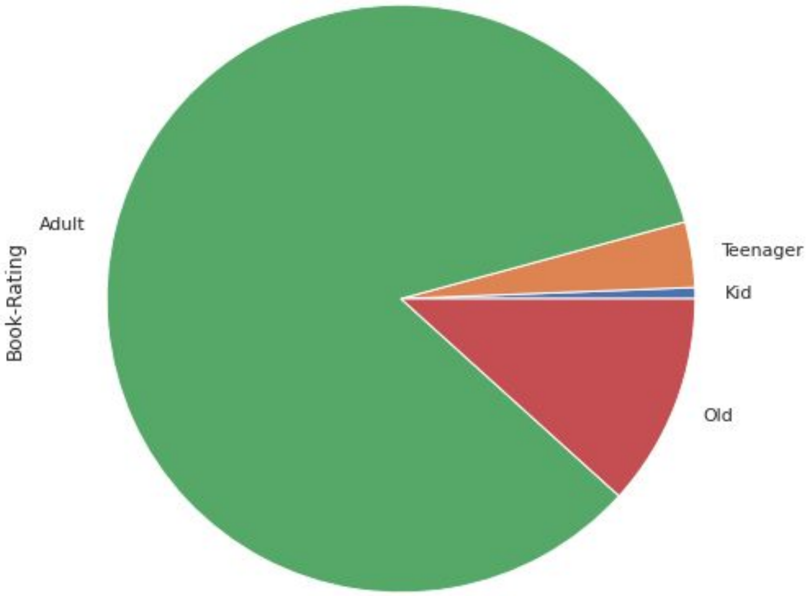

## ➢ Insights

- **Here, we are able to observe, which age bin has contributed most to the Book-Ratings**

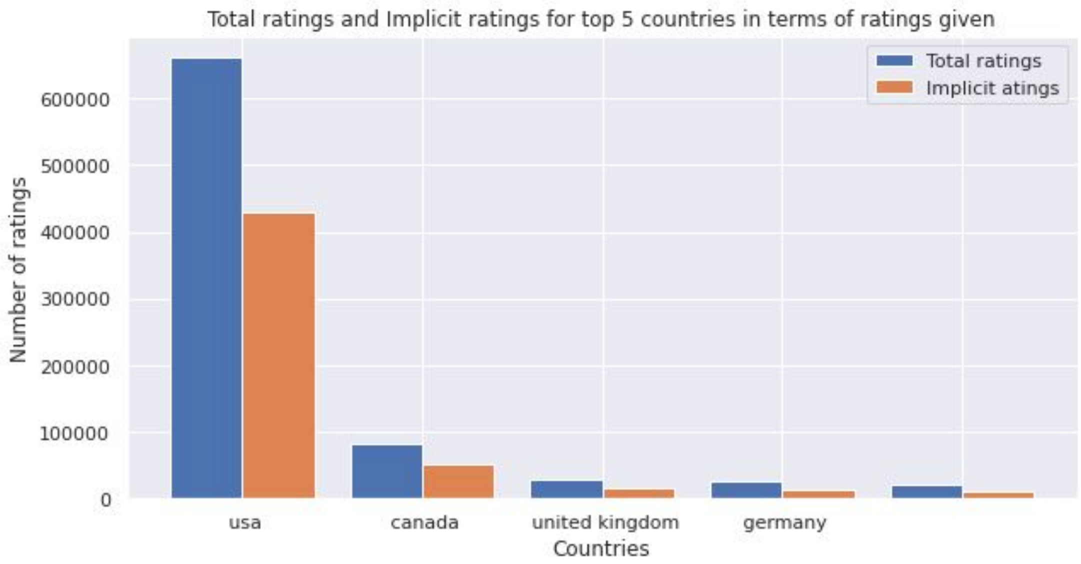
# Primary EDA: Age Bin Representation in the Dataset

## ➢ Insights

- Most customers are Adults (20-50yrs)
- 2nd most represented age group is for boomers (>50yrs)



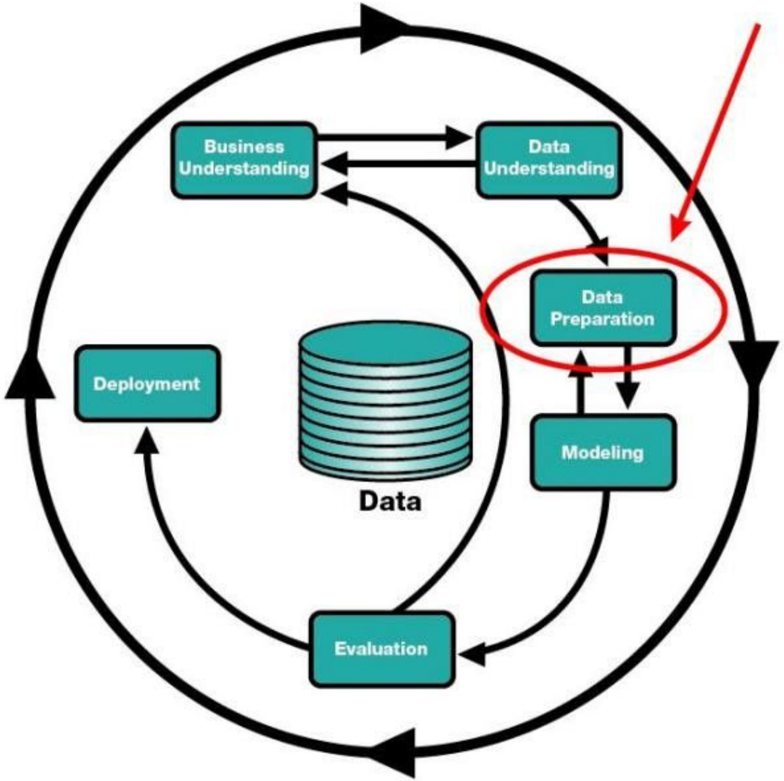
# Primary EDA: Implicit Ratings, as a fraction of Total Ratings, per country




## ➢ Insights

- We can see, implicit ratings appear as a fraction of the total ratings in a similar ratio across all countries

# Preparing the Data For Models



# Preparing Data for Model

AI

- **Cleaning Year of Publication:** It was observed that there is noise in the Year of Publication features :-
  - String Noise Values - such as 'DK Publishing Inc' and 'Gallimard.
  - Integer Noise Values - Since this data was collected in August 2006, so any year value greater than 2006 is a noise value.
  - Therefore, after cleaning the dataset based upon Year-Of-Publication Feature, we lost only a miniscule amount of 1.3% data.

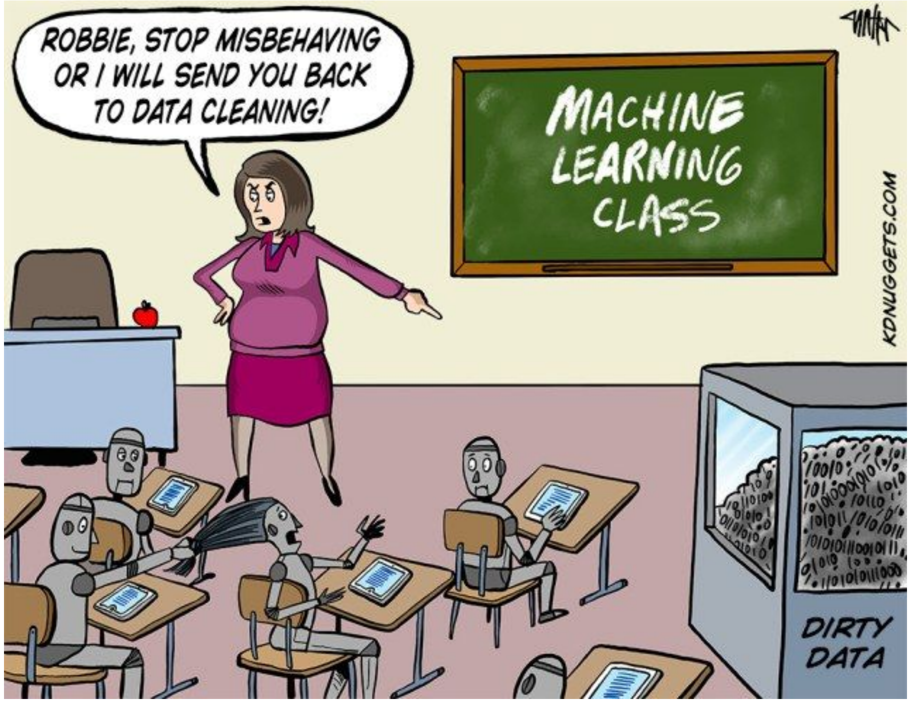
- **Selecting Books with Optimum Number of Ratings:** Building a recommendation system requires a lot of data. Recommendations should be relevant, otherwise they can cause a nuisance to the customers. So, we have set a threshold number of ratings per book in order to get optimal recommendations for our users.

# ➢ Preparing Data for Model

AI

- **Defining Optimum Reader:** We can't take every user's rating at face value because if the user is a novice reader with only an experience of reading a couple of books, his/her ratings might not be much relevant for finding similarity among books. Therefore, as a general rule of thumb, we're choosing only those Users who have rated at least 10 Books for building the recommendation system

# Applying Models



# Applying Model: Recommendations Based upon Top 10 Books with the highest average rating (Explicit)

| Book-Title | Rating-Mean |
|---|---|
| The Baby Book: Everything You Need to Know About Your Baby from Birth to Age Two | 8.46 |
| Die unendliche Geschichte: Von A bis Z | 8.07 |
| Free | 8.02 |
| There's Treasure Everywhere--A Calvin and Hobbes Collection | 7.88 |
| Harry Potter y el cÃ¡liz de fuego | 7.88 |
| Warchild | 7.62 |
| Jesus Freaks: DC Talk and The Voice of the Martyrs - Stories of Those Who Stood For Jesus, the Ultimate Jesus Freaks | 7.53 |
| El Hobbit | 7.48 |
| A Night Without Armor : Poems | 7.25 |
| The Napping House | 7.21 |

# **Applying Model:** Interactive: Top 10 books for respective authors - Let's Head to a short Demo (Explicit)

Book_Author: Nora Roberts

| Book-Title | Rating-Mean |
|---|---|
| Hidden Star (The Stars Of Mithra) (Harlequin Intimate Moments, 811) | 5.00 |
| Secret Star (The Stars Of Mithra) (Harlequin Silhouette Intimate Moments, No 835) | 4.44 |
| Entranced | 4.22 |
| Love By Design | 3.89 |
| Going Home: Unfinished Business/ Island of Flowers/ Mind Over Matter | 3.79 |
| Captivated (Silhouette Single Title) | 3.78 |
| Midnight Bayou | 3.76 |
| Chesapeake Blue (Quinn Brothers (Hardcover)) | 3.69 |
| Last Honest Woman | 3.58 |
| Key of Valor (Roberts, Nora. Key Trilogy, 3.) | 3.56 |

# Applying Model: Memory Based KNN Model (Explicit)

AI

**A KNN model, with cosine similarity as a metric for measuring the distance between different ratings, was used to provide recommendations**

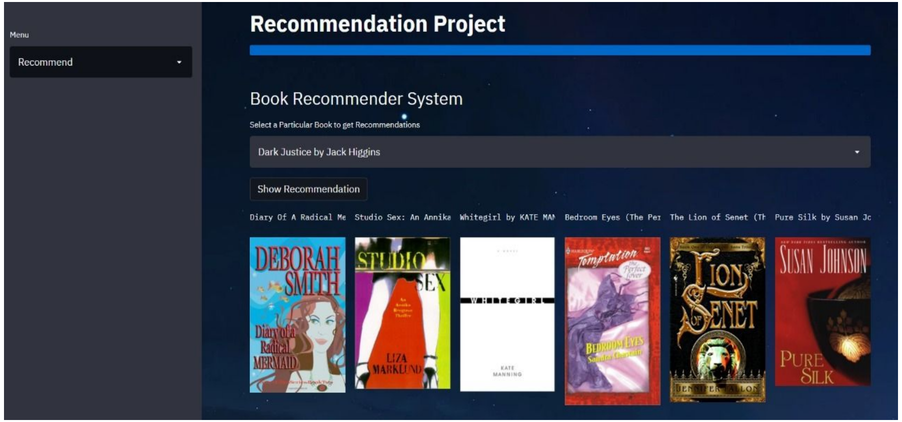
```
recommend('9-11 by Noam Chomsky', n_values=10)

The Top 9 Recommendations for Users who have read book 9-11 by Noam Chomsky are shown below:-
1: Die Weiss Lowin / Contemporary German Lit by Henning Mankell, with distance of 0.6220355269907728.
2: The First Counsel by Brad Meltzer, with distance of 0.6220355269907728.
3: Schlafes Bruder by Robert Schneider, with distance of 0.6220355269907728.
4: Herzsprung by Ildiko Kurthy, with distance of 0.6220355269907728.
5: Due di due (Bestsellers) by Andrea De Carlo, with distance of 0.6220355269907728.
6: MÃ?Â¶rder ohne Gesicht. by Henning Mankell, with distance of 0.6220355269907728.
7: UN Viejo Que Leia Novelas De Amor/the Old Men Who Read Love Stories (ColecciÃ³n Andanzas) by Luis Sepulveda, with distance of 0.6220355269907728.
8: Vernon God Little: A 21st Century Comedy in the Presence of Death by D. B. C. Pierre, with distance of 0.6220355269907728.
9: Lauf, Jane, lauf. Roman. by Joy Fielding, with distance of 0.6220355269907728.
```

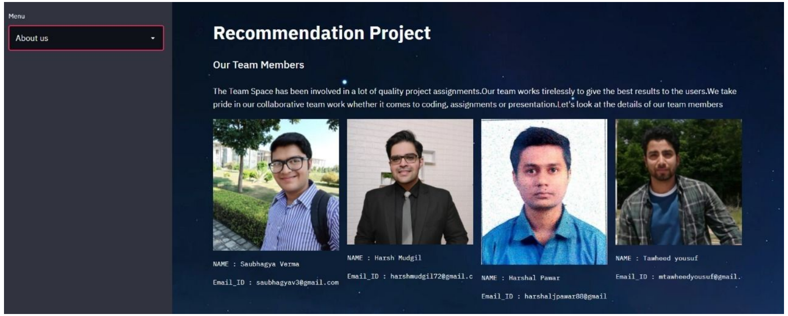
## ➢ Insight

**We can see, that the recommended books, are quite similar in genre to the selected item**

# KNN Model in Action on Website




Making Book Recommendation




About Us

# Applying Model: Collaborative Filtering Model (Explicit)

**It makes predictions about the interests of a user by collecting preferences from many users. The underlying assumption is, if a person A has the same opinion as a person B on a set of items, A is more likely to have B's opinion for a given item than that of a randomly chosen person.**

```
Global metrics:
{'modelName': 'Collaborative Filtering', 'recall@5': 0.9977659057795046, 'recall@10': 0.9977659057795046}
```

| | hits@5_count | hits@10_count | interacted_count | recall@5 | recall@10 | _person_id |
|---|---|---|---|---|---|---|
| **12** | 279 | 279 | 289 | 0.965398 | 0.965398 | 11676 |
| **31** | 51 | 51 | 51 | 1.000000 | 1.000000 | 16795 |
| **25** | 48 | 48 | 48 | 1.000000 | 1.000000 | 104636 |
| **41** | 48 | 48 | 48 | 1.000000 | 1.000000 | 153662 |
| **339** | 47 | 47 | 47 | 1.000000 | 1.000000 | 95359 |
| **241** | 47 | 47 | 47 | 1.000000 | 1.000000 | 98391 |
| **232** | 44 | 44 | 44 | 1.000000 | 1.000000 | 114368 |
| **510** | 35 | 35 | 35 | 1.000000 | 1.000000 | 123883 |
| **216** | 33 | 33 | 33 | 1.000000 | 1.000000 | 60244 |
| **464** | 31 | 31 | 31 | 1.000000 | 1.000000 | 158295 |

# Applying Model: Collaborative Filtering Model (Explicit)

**We can see, the user: 40943, has rated Harry Potter and the Sorcerer's Stone (Book 1), very highly. Our model, is recommending other parts of the same series. This seems to be consistent with high precision and high recall values that we have obtained thus far.**

| | User-ID | ISBN | Book-Rating | Book-Title |
|---|---|---|---|---|
| **367478** | 40943 | 0671003755 | 5 | She's Come Undone (Oprah's Book Club (Paperback)) |
| **367497** | 40943 | 0679746048 | 8 | Girl, Interrupted |
| **367499** | 40943 | 039480967X | 5 | Bears on Wheels (Bright & Early Books) |
| **367514** | 40943 | 043936213X | 10 | Harry Potter and the Sorcerer's Stone (Book 1) |
| **367518** | 40943 | 0553274295 | 10 | Where the Red Fern Grows |

```
] recc
```

```
array(['Harry Potter and the Chamber of Secrets (Book 2)',
       'Harry Potter and the Prisoner of Azkaban (Book 3)',
       'Harry Potter and the Goblet of Fire (Book 4)',
       "Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))",
       'The Secret Life of Bees',
       'Harry Potter and the Order of the Phoenix (Book 5)',
       "Harry Potter and the Sorcerer's Stone (Book 1)",
       "Bridget Jones's Diary",
       'The Fellowship of the Ring (The Lord of the Rings, Part 1)',
       'The Nanny Diaries: A Novel'], dtype=object)
```

# Applying Model: Recommendations for implicit case

## With KNN Based Model

- In case of implicit ratings,  we can not know the exact preferences of a user regarding these books.
- For such users, we have decided to build a recommendation system using the criteria of Age as relevance for the recommendations.

```
get_recommendations(508,10)

{'A Painted House',
 'Cat &amp; Mouse (Alex Cross Novels)',
 'False Memory',
 'Irish Hearts',
 'Night Prey',
 'Red Dragon',
 "Songs in Ordinary Time (Oprah's Book Club (Paperback))",
 'Stone Kiss',
 'The Beach House',
 'We Were the Mulvaneys'}
```

```
get_recommendations(677,10)

{'A Painted House by John Grisham',
 'False Memory by Dean R. Koontz',
 'Mr. Murder by Dean R. Koontz',
 'Sea Glass: A Novel by Anita Shreve',
 'Shopaholic Takes Manhattan (Summer Display Opportunity) by Sophie Kinsella',
 "Songs in Ordinary Time (Oprah's Book Club (Paperback)) by Mary McGarry Morris",
 'Still Waters by TAMI HOAG',
 'Stone Kiss by Faye Kellerman',
 'The Beach House by James Patterson',
 'The Notebook by Nicholas Sparks'}
```

# Applying Model: Content Description Based Recommender system

- **Fetching Descriptions:** Descriptions for Book-Titles were fetched from Google Books API. Restrictions on the number of requests at a time, were circumvented using Python snippet.

- **Data Preprocessing:** Stop words were removed from descriptions. The data was the Stemmed. Finally, data was vectorized using TFIDF vectorizer.

- **Measuring Similarity:** Similarity of different documents was measured, based on cosine-similarity metric. Recommendations were made using ranked matrix.

# Applying Model: Content Description Based Recommender system

```
#Recommendations based on -> 193156146X: The Time Traveler's Wife
recommendations('193156146X')

['Florida Roadkill',
 'Bleachy-Haired Honky Bitch : Tales from a Bad Neighborhood',
 'Miss Julia Takes over',
 'The Unlikely Ones',
 "Schindler's List",
 'The Passion of Artemisia',
 'Confessions of a Sociopathic Social Climber : The Katya Livingston Chronicles (Katya Livingston Chronicles (Hardcover))',
 "Murphy's Law (A Mitch Mitchell Mystery)",
 "Pigs Don't Fly",
 'The Other Boleyn Girl']

#Similar Recommendations:
[i for i in recommendations('0312968884')  if i in recommendations('193156146X')]

["Murphy's Law (A Mitch Mitchell Mystery)"]

#Recommendations based on -> 0688167829: Florida Roadkill
recommendations('0688167829')

['Coraline',
 'Scales of Justice (Inspector Roderick Alleyn Mysteries)',
 'The Big Bounce',
 'Love in the Time of Cholera (Penguin Great Books of the 20th Century)',
 'Confessions of a Sociopathic Social Climber : The Katya Livingston Chronicles (Katya Livingston Chronicles (Hardcover))',
 'Everville : The Second Book of the Art',
 'Murder on a Bad Hair Day: A Southern Sisters Mystery',
 "Murphy's Law (A Mitch Mitchell Mystery)",
 'The Inn at Lake Devine',
 'Galilee']

#Similar Recommendations:
[i for i in recommendations('193156146X')  if i in recommendations('0688167829')]

['Confessions of a Sociopathic Social Climber : The Katya Livingston Chronicles (Katya Livingston Chronicles (Hardcover))',
 "Murphy's Law (A Mitch Mitchell Mystery)"]
```

**We can notice that, books recommended for a user who has previously read 'The Time Traveller's Wife', have respective recommendations which are similar to each other. Thus, the content similarly based recommendation system is working as, expected.**

# Conclusion

- It is very important to deal with implicit and explicit user ratings, separately.

- For dealing with explicit ratings, we can build simple models based on ratings, and we can also use certain comprehensive models based on Collaborative Filtering approach.

- For dealing with implicit ratings, we can build KNN based models , and we can also use content based models, which utilize the similarity of different contents, to make recommendations

- It is crucial to be precise about user preferences, otherwise repetitive recommendations can cause nuisance to the user.

# Challenges

- Dealing and filtering data, to reach to the most recommendable users was an adventurous task to do
- Collating the analysis of different team members, was a difficult task.
- Exploring literature and resources to understand the problem and to find the solution was a little exhaustive.
- Deadlines felt a little strained. But it all worked out for the best.

# QnA