# Capstone Project-4

## Book Recommendation System

**Member**

**Sourabh Pramanik**

# Contents


1. Problem Statement
2. Data Summary
3. Checking Missing values
4. EDA and Visualization
5. Applying  Machine Learning model
6. Output
7. Final Summary of Conclusion
8. Q&A

# Problem Statement

Here in this project we have three dataset-

1. Books
2. Users
3. Ratings

In today's world there are lots of books present. Users read books and give ratings and reviews according to their interest and experience.

In this project we will create a recommendation system which will provide genuine suggestion of the books to the user.

# Data Summary

We will complete this project by using following steps-

- After reading the data we will perform Exploratory Data analysis.
- We will check the Null values .
- We will do some Visualization.
- After then we will apply Machine Learning model for Recommendation.

# Data Summary

Outcome of this Project -

- Recommendation of best books matching with user's interest.

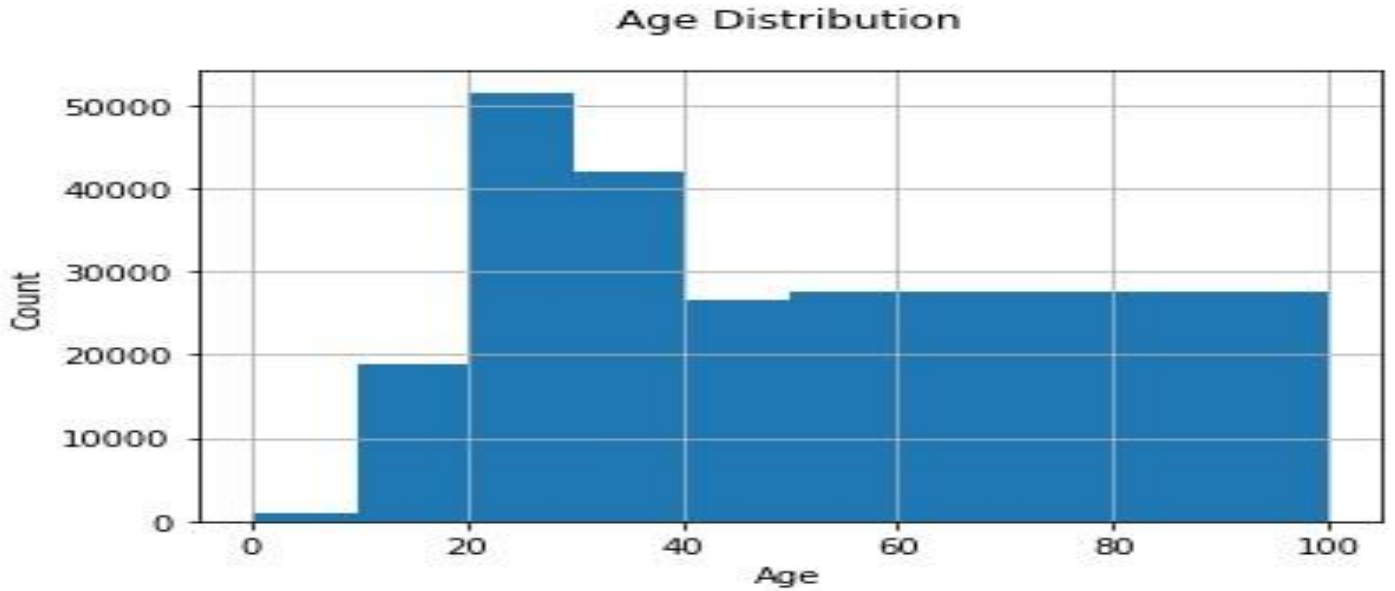
# Checking Missing values

## Missing Values

```
users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278858 entries, 0 to 278857
Data columns (total 3 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   userID    278858 non-null  int64
 1   Location  278858 non-null  object
 2   Age       168096 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 6.4+ MB
```

# EDA and Visualization



# EDA and Visualization

```
ratings['userID'].value_counts()
```

```
11676     13602
198711     7550
153662     6109
98391      5891
35859      5850
          ...
158698        1
17920         1
277135        1
275086        1
187812        1
Name: userID, Length: 105283, dtype: int64
```

**Now we will exclude some records for better Recommendation-**

1. **Users with less than 200 Ratings given.**
2. **Books with less than 100 Ratings.**

# EDA and Visualization

```
popularity_threshold = 50                                                    #Threshold value
rating_popular_book = rating_with_totalRatingCount.query('totalRatingCount >= @popularity_threshold')

rating_popular_book.head()
```

| | userID | ISBN | bookRating | bookTitle | totalRatingCount |
|---|---|---|---|---|---|
| 0 | 277427 | 002542730X | 10 | Politically Correct Bedtime Stories: Modern Ta... | 82 |
| 1 | 3363 | 002542730X | 0 | Politically Correct Bedtime Stories: Modern Ta... | 82 |
| 2 | 11676 | 002542730X | 6 | Politically Correct Bedtime Stories: Modern Ta... | 82 |
| 3 | 12538 | 002542730X | 10 | Politically Correct Bedtime Stories: Modern Ta... | 82 |
| 4 | 13552 | 002542730X | 0 | Politically Correct Bedtime Stories: Modern Ta... | 82 |

# EDA and Visualization

**As we have data of different different regions so we will filter the data of US and Canada**

| | userID | ISBN | bookRating | bookTitle | totalRatingCount | Location |
|---|---|---|---|---|---|---|
| 0 | 277427 | 002542730X | 10 | Politically Correct Bedtime Stories: Modern Ta... | 82 | gilbert, arizona, usa |
| 1 | 3363 | 002542730X | 0 | Politically Correct Bedtime Stories: Modern Ta... | 82 | knoxville, tennessee, usa |
| 3 | 12538 | 002542730X | 10 | Politically Correct Bedtime Stories: Modern Ta... | 82 | byron, minnesota, usa |
| 4 | 13552 | 002542730X | 0 | Politically Correct Bedtime Stories: Modern Ta... | 82 | cordova, tennessee, usa |
| 5 | 16795 | 002542730X | 0 | Politically Correct Bedtime Stories: Modern Ta... | 82 | mechanicsville, maryland, usa |

# Modelling

| userID | 254 | 2276 | 2766 | 2977 | 3363 | 4017 | 4385 | 6242 | 6251 | 6323 | 6543 | 6563 | 6575 | 7158 | 7286 | 7346 | 7915 | 8067 | 8245 | 8681 | 8936 | 9856 | 10447 | 10819 | 11601 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bookTitle | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1984 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 1st to Die: A Novel | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 2nd Chance | 0.0 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 4 Blondes | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | |
| 84 Charing Cross Road | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

# Modelling

```
from sklearn.neighbors import NearestNeighbors

model_knn = NearestNeighbors(metric ='cosine', algorithm ='brute')

model_knn.fit(us_canada_user_rating_matrix)
```

```
NearestNeighbors(algorithm='brute', leaf_size=30, metric='cosine',
                 metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                 radius=1.0)
```

# Output

```
Recommendations for Red Dragon:

1: Hannibal, with distance of 0.646341068098635:
2: Silence of the Lambs, with distance of 0.7205469453143072:
3: Jurassic Park, with distance of 0.7612529735136639:
4: Red Storm Rising, with distance of 0.7682899189051628:
5: Servant of the Bones, with distance of 0.7726643703980889:
```

# Conclusion

**The Nearest Neighbour Algorithm working well with the Book data.**

**We can use the same concept in Movies Recommendation, Videos Recommendation for suggestion according to customer's interest.**

**Here we can also use content based filtering.**

# References


https://www.analyticsvidhya.com/

https://towardsdatascience.com/

https://stackoverflow.com/

# Q & A