# Book Recommendation System

**Jayalaxmi Mekap**

**Data science trainees,**

**AlmaBetter, Bangalore**

## Abstract:

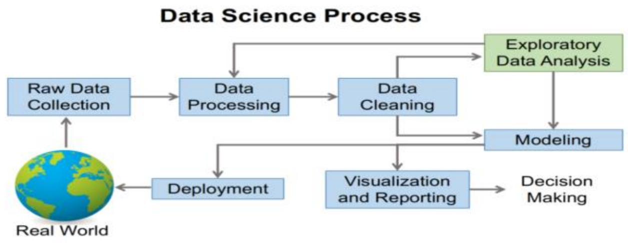
Recommendation systems is used for the purpose of suggesting items to purchase or to see. They direct users towards those items which can meet their needs through cutting down large database of Information. A various techniques have been introduced for recommending items i.e., content, collaborative and association mining techniques are used. This paper solves the problem of data sparsity problem by combining the collaborative-based filtering and association rule mining to achieve better performance. The results obtained are demonstrated and the proposed recommendation algorithms perform better and solve the challenges such as data sparsity and scalability.

## Objective:

The main objective is to create a book recommendation system for users. 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.

Steps involve in this project

- loading the data into data frame
- cleaning the data
- extracting statistics from the dataset
- exploratory analysis and visualizations
- questions that can be asked from the dataset
- conclusion



## 1.Problem Statement

• During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyer's articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems aretoday unavoidable in our daily online journeys.

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy, or anything else

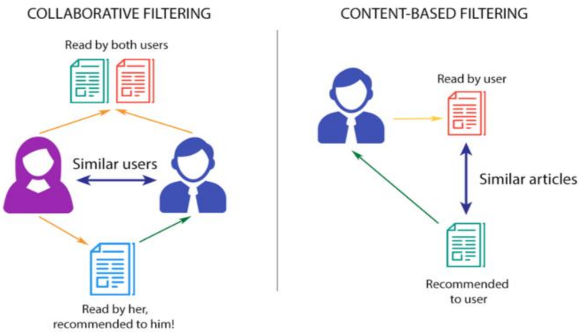
depending on industries).

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. The main objective is to create a book recommendation system for users.

## 2. Introduction

We were provided the in-CSV format there are 3 data sets names books, users and ratings -

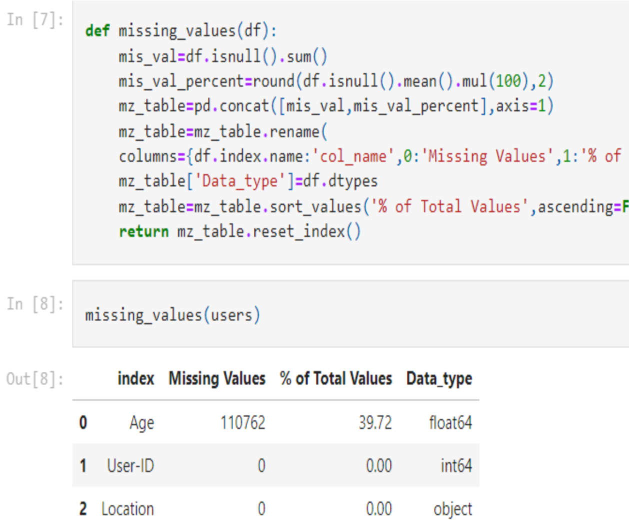
- Users: Contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.
- Books: Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon website.
- Ratings: Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.



## 3. Data Cleanings and validations

In this step removing faulty data and filling in gaps.  The task to be crucial and important thus validating by following steps

- Removing extraneous data
- Handing in missing values.
- Data shifting in respective columns
- Conforming data to a standardized pattern.
-

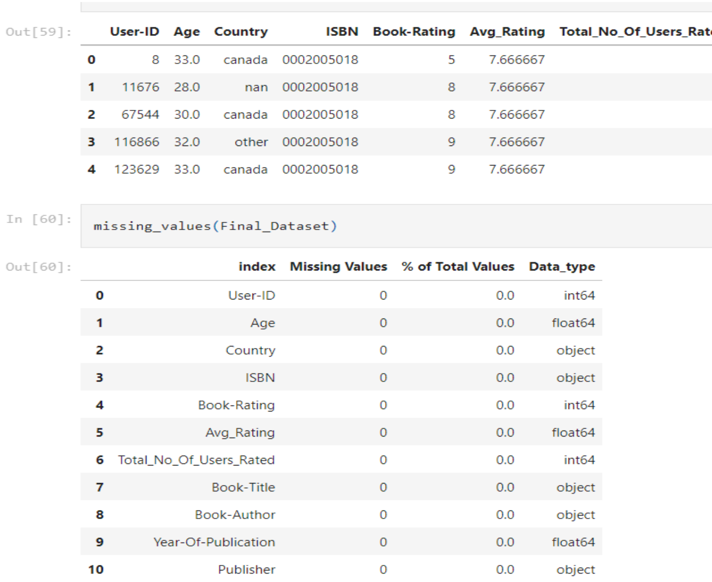


In [7]:
```
def missing_values(df):
    mis_val=df.isnull().sum()
    mis_val_percent=round(df.isnull().mean().mul(100),2)
    mz_table=pd.concat([mis_val,mis_val_percent],axis=1)
    mz_table=mz_table.rename(
    columns={df.index.name:'col_name',0:'Missing Values',1:'% of
    mz_table['Data_type']=df.dtypes
    mz_table=mz_table.sort_values('% of Total Values',ascending=F
    return mz_table.reset_index()
```

In [8]:
```
missing_values(users)
```

Out[8]:

| | index | Missing Values | % of Total Values | Data_type |
|---|---|---|---|---|
| 0 | Age | 110762 | 39.72 | float64 |
| 1 | User-ID | 0 | 0.00 | int64 |
| 2 | Location | 0 | 0.00 | object |



Out[59]:

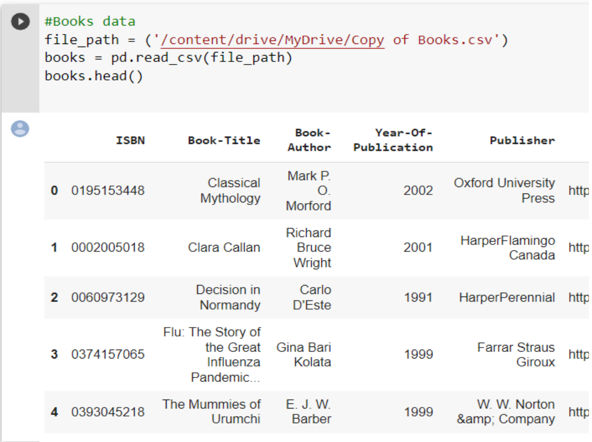
| | User-ID | Age | Country | ISBN | Book-Rating | Avg_Rating | Total_No_Of_Users_Rat |
|---|---|---|---|---|---|---|---|
| 0 | 8 | 33.0 | canada | 0002005018 | 5 | 7.666667 | |
| 1 | 11676 | 28.0 | nan | 0002005018 | 8 | 7.666667 | |
| 2 | 67544 | 30.0 | canada | 0002005018 | 8 | 7.666667 | |
| 3 | 116866 | 32.0 | other | 0002005018 | 9 | 7.666667 | |
| 4 | 123629 | 33.0 | canada | 0002005018 | 9 | 7.666667 | |

In [60]:
```
missing_values(Final_Dataset)
```

Out[60]:

| | index | Missing Values | % of Total Values | Data_type |
|---|---|---|---|---|
| 0 | User-ID | 0 | 0.0 | int64 |
| 1 | Age | 0 | 0.0 | float64 |
| 2 | Country | 0 | 0.0 | object |
| 3 | ISBN | 0 | 0.0 | object |
| 4 | Book-Rating | 0 | 0.0 | int64 |
| 5 | Avg_Rating | 0 | 0.0 | float64 |
| 6 | Total_No_Of_Users_Rated | 0 | 0.0 | int64 |
| 7 | Book-Title | 0 | 0.0 | object |
| 8 | Book-Author | 0 | 0.0 | object |
| 9 | Year-Of-Publication | 0 | 0.0 | float64 |
| 10 | Publisher | 0 | 0.0 | object |

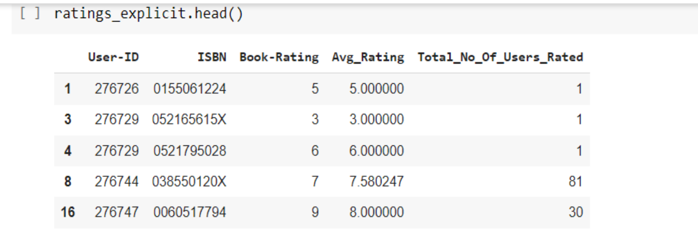
## 4. Transform data



```
#Books data
file_path = ('/content/drive/MyDrive/Copy of Books.csv')
books = pd.read_csv(file_path)
books.head()
```

| | ISBN | Book-Title | Book-Author | Year-Of-Publication | Publisher | |
|---|---|---|---|---|---|---|
| 0 | 0195153448 | Classical Mythology | Mark P. O. Morford | 2002 | Oxford University Press | http |
| 1 | 0002005018 | Clara Callan | Richard Bruce Wright | 2001 | HarperFlamingo Canada | http |
| 2 | 0060973129 | Decision in Normandy | Carlo D'Este | 1991 | HarperPerennial | http |
| 3 | 0374157065 | Flu: The Story of the Great Influenza Pandemic... | Gina Bari Kolata | 1999 | Farrar Straus Giroux | http |
| 4 | 0393045218 | The Mummies of Urumchi | E. J. W. Barber | 1999 | W. W. Norton &amp; Company | http |

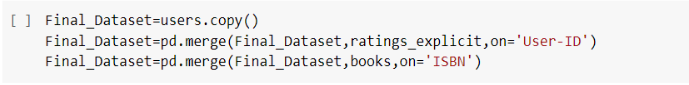
●



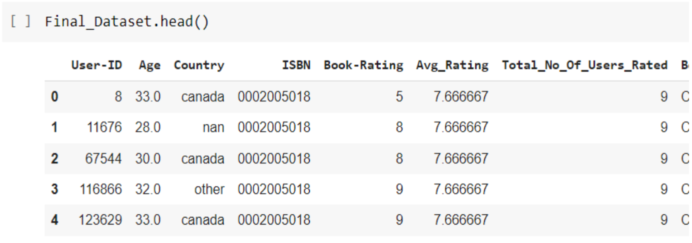
```
[ ] ratings_explicit.head()
```

| | User-ID | ISBN | Book-Rating | Avg_Rating | Total_No_Of_Users_Rated |
|---|---|---|---|---|---|
| 1 | 276726 | 0155061224 | 5 | 5.000000 | 1 |
| 3 | 276729 | 052165615X | 3 | 3.000000 | 1 |
| 4 | 276729 | 0521795028 | 6 | 6.000000 | 1 |
| 8 | 276744 | 038550120X | 7 | 7.580247 | 81 |
| 16 | 276747 | 0060517794 | 9 | 8.000000 | 30 |

### Merging All Dataset.

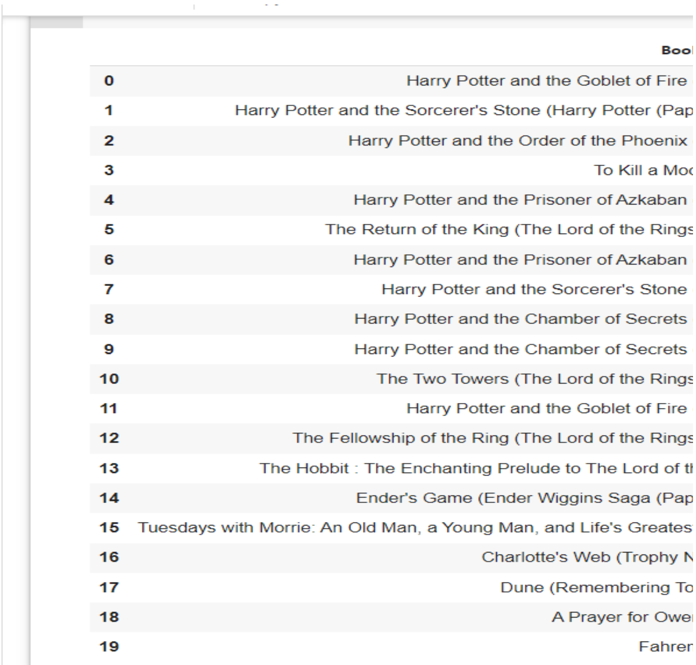


```
[ ] Final_Dataset=users.copy()
    Final_Dataset=pd.merge(Final_Dataset,ratings_explicit,on='User-ID')
    Final_Dataset=pd.merge(Final_Dataset,books,on='ISBN')
```

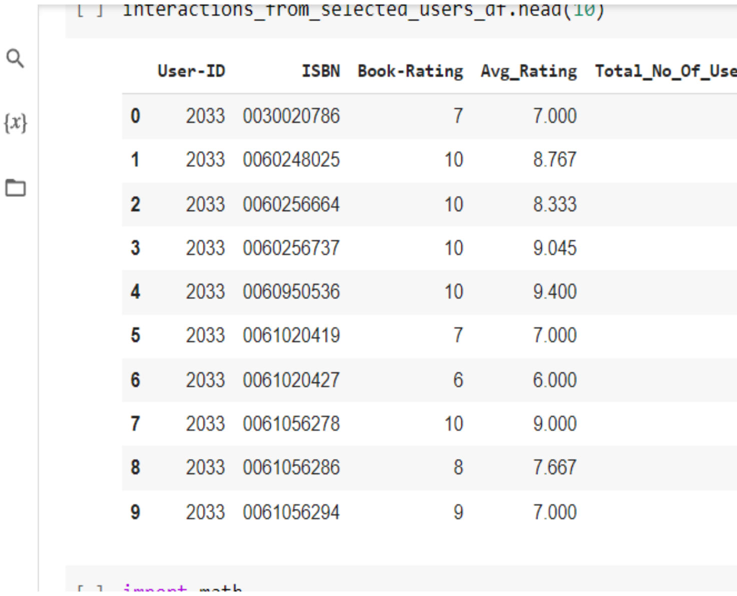


```
[ ] Final_Dataset.head()
```

| | User-ID | Age | Country | ISBN | Book-Rating | Avg_Rating | Total_No_Of_Users_Rated | B |
|---|---|---|---|---|---|---|---|---|
| 0 | 8 | 33.0 | canada | 0002005018 | 5 | 7.666667 | 9 | C |
| 1 | 11676 | 28.0 | nan | 0002005018 | 8 | 7.666667 | 9 | C |
| 2 | 67544 | 30.0 | canada | 0002005018 | 8 | 7.666667 | 9 | C |
| 3 | 116866 | 32.0 | other | 0002005018 | 9 | 7.666667 | 9 | C |
| 4 | 123629 | 33.0 | canada | 0002005018 | 9 | 7.666667 | 9 | C |



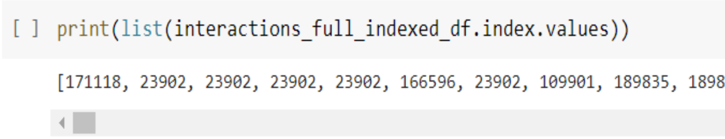
| | Boo |
|---|---|
| 0 | Harry Potter and the Goblet of Fire |
| 1 | Harry Potter and the Sorcerer's Stone (Harry Potter (Pap |
| 2 | Harry Potter and the Order of the Phoenix |
| 3 | To Kill a Moc |
| 4 | Harry Potter and the Prisoner of Azkaban |
| 5 | The Return of the King (The Lord of the Rings |
| 6 | Harry Potter and the Prisoner of Azkaban |
| 7 | Harry Potter and the Sorcerer's Stone |
| 8 | Harry Potter and the Chamber of Secrets |
| 9 | Harry Potter and the Chamber of Secrets |
| 10 | The Two Towers (The Lord of the Rings |
| 11 | Harry Potter and the Goblet of Fire |
| 12 | The Fellowship of the Ring (The Lord of the Rings |
| 13 | The Hobbit : The Enchanting Prelude to The Lord of tl |
| 14 | Ender's Game (Ender Wiggins Saga (Pap |
| 15 | Tuesdays with Morrie: An Old Man, a Young Man, and Life's Greates |
| 16 | Charlotte's Web (Trophy N |
| 17 | Dune (Remembering To |
| 18 | A Prayer for Owe |
| 19 | Fahrer |



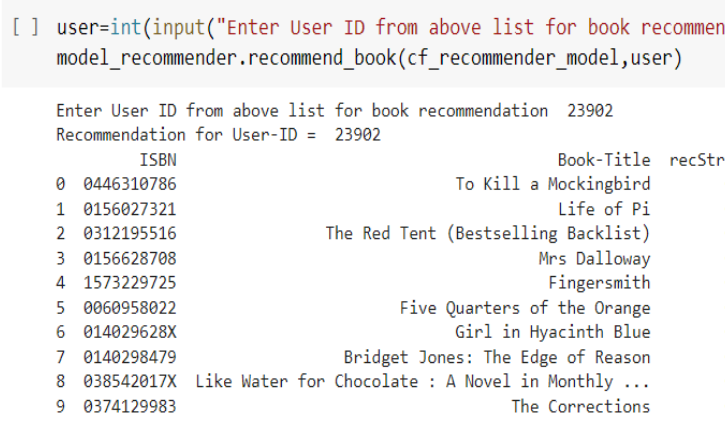
```
[ ] interactions_from_selected_users_df.head(10)
```

| | User-ID | ISBN | Book-Rating | Avg_Rating | Total_No_Of_Use |
|---|---|---|---|---|---|
| 0 | 2033 | 0030020786 | 7 | 7.000 | |
| 1 | 2033 | 0060248025 | 10 | 8.767 | |
| 2 | 2033 | 0060256664 | 10 | 8.333 | |
| 3 | 2033 | 0060256737 | 10 | 9.045 | |
| 4 | 2033 | 0060950536 | 10 | 9.400 | |
| 5 | 2033 | 0061020419 | 7 | 7.000 | |
| 6 | 2033 | 0061020427 | 6 | 6.000 | |
| 7 | 2033 | 0061056278 | 10 | 9.000 | |
| 8 | 2033 | 0061056286 | 8 | 7.667 | |
| 9 | 2033 | 0061056294 | 9 | 7.000 | |

```
[ ] import math
```

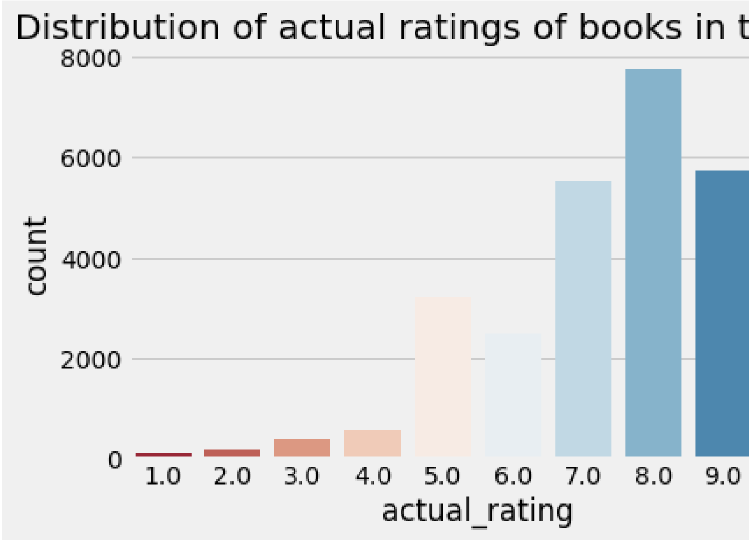


```
[ ] print(list(interactions_full_indexed_df.index.values))

[171118, 23902, 23902, 23902, 23902, 166596, 23902, 109901, 189835, 1898
```

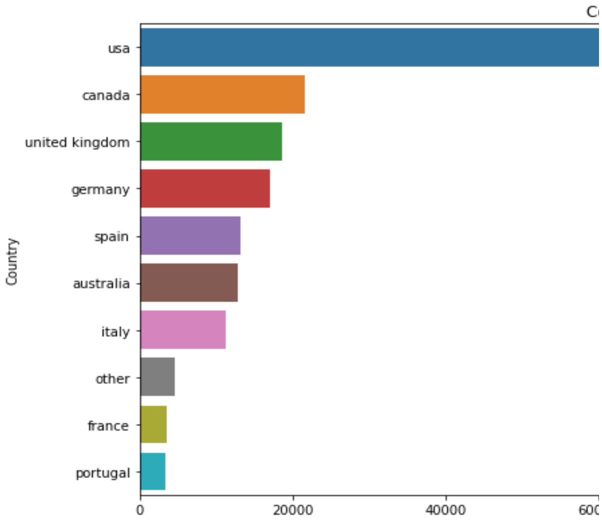


```
[ ] user=int(input("Enter User ID from above list for book recommen
    model_recommender.recommend_book(cf_recommender_model,user)

Enter User ID from above list for book recommendation  23902
Recommendation for User-ID =  23902
         ISBN                                     Book-Title   recStr
0  0446310786                           To Kill a Mockingbird
1  0156027321                                      Life of Pi
2  0312195516             The Red Tent (Bestselling Backlist)
3  0156628708                                    Mrs Dalloway
4  1573229725                                     Fingersmith
5  0060958022                    Five Quarters of the Orange
6  014029628X                          Girl in Hyacinth Blue
7  0140298479               Bridget Jones: The Edge of Reason
8  038542017X  Like Water for Chocolate : A Novel in Monthly ...
9  0374129983                                The Corrections
```

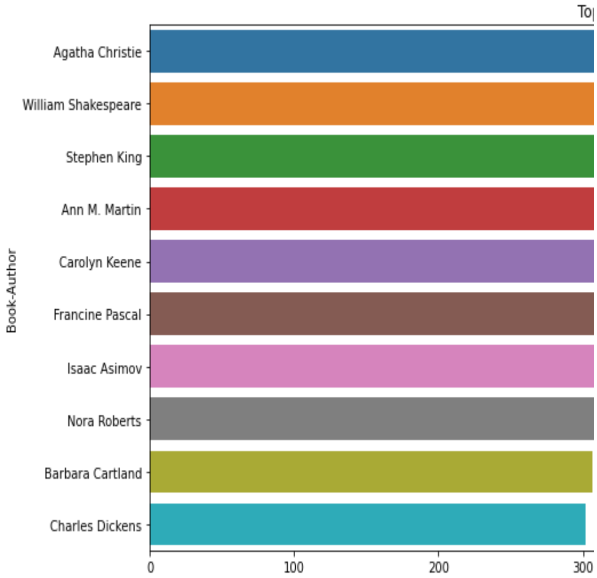
# 5. Exploratory Analysis and Visualization



- Country wise Users who read books



. Top 10 Authors



## 6. Contributions:

1. **Introduction to data**
2. **Problem statement**
3. **Data Summary**
4. **Analysis of different datasets**
5. **Data Cleaning and pre processing**
6. **Understanding the data**
7. **correlation**
8. **Outlier treatment**
9. **Imputing missing values**
10. **Implanting algorithms**
11. **Popularity based recommendations**
12. **Popular in whole collection**
13. **Popular in given place**
14. **Books by same author, publisher of given book name**
15. **Popular books yearly**
16. **Recommendation using average weighted rating**
17. **Different Recommendation Model**
18. **User item collaborative filtering recommendation**
19. **Correlation based recommendation**
20. **Nearest neighbors-based recommendation**
21. **Content based recommendation**
22. **Challenges**
23. **Conclusion**
24. **Future Scope**

## 7. Programing Language:

We have used python programing Language and used below library for EDA

Numpy

Pandas

Seaborn

maths

matplotlib

warnings