Book Recommendation System

Submitted in partial fulfillment of the requirements
of the degree of
Bachelor of Computer Engineering

B. E. Computer Engineering

By

Shravani Dhuri	**Roll No. 06**	**PID: 182026**
Clare Rebello	**Roll No. 07**	**PID: 128092**
Kate Rebello	**Roll No. 08**	**PID: 182093**
Janhvi Shah	**Roll No. 13**	**PID: 182104**

Guide:
Mrs. G. ANURADHA
Associate Professor

Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)

University of Mumbai

2021-2022

Content:

Problem Statement:

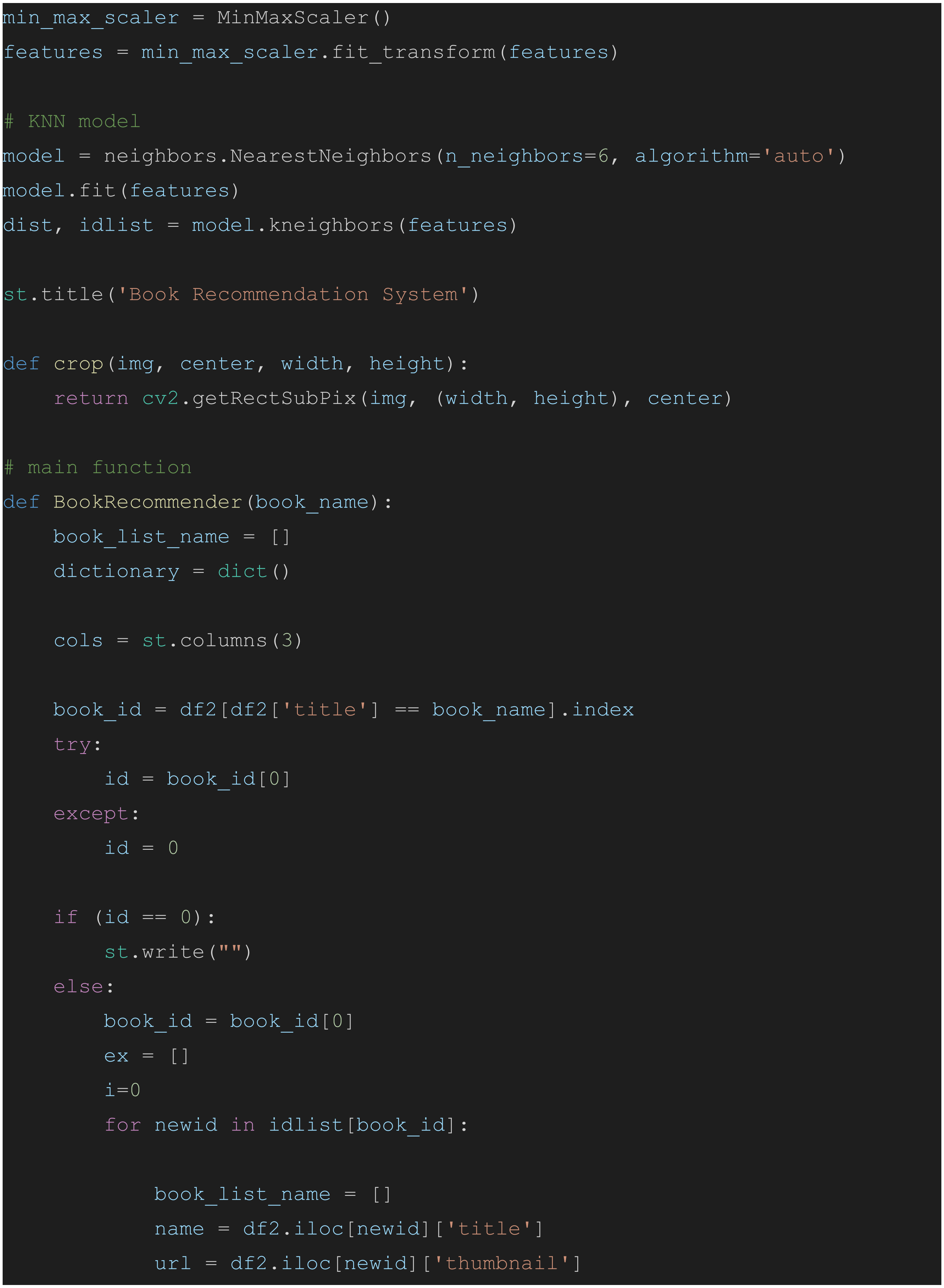
In many applications today, Recommender systems usually provide the user with a list of recommendations that they might prefer, or supply predictions on how much the user might prefer each item.The purpose of a book recommendation system is to predict buyer’s interest and recommend books to them accordingly. A book recommendation system can take into account many parameters like book content and book quality by filtering user reviews.

Objectives:

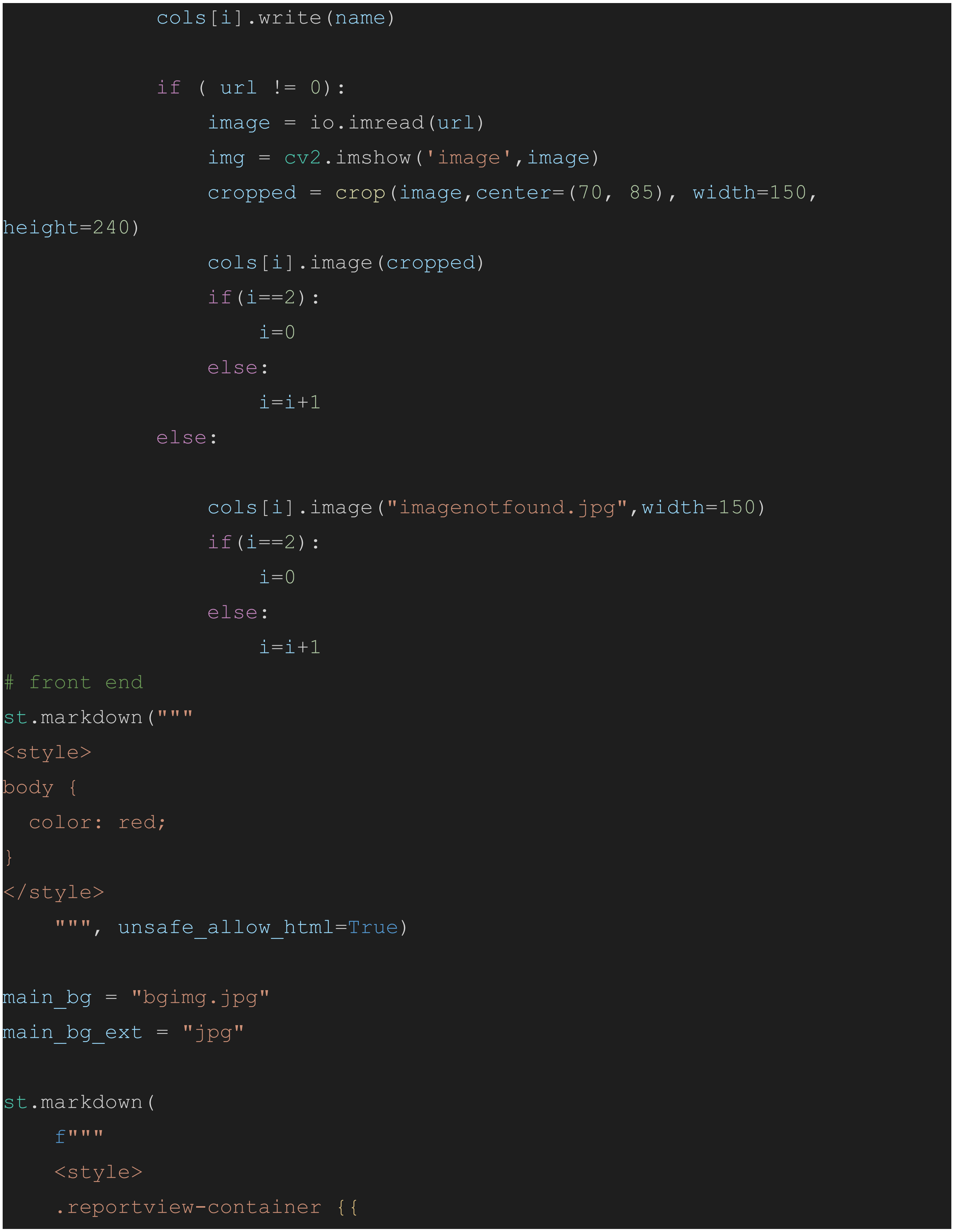
Recommendation systems are widely used today to recommend products to users based on their interests. A recommendation system is one of the strongest systems for increasing profits by retaining more users in a very big competition.A recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. A recommendation engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behaviour of a customer and based on that, recommends products which the users might be likely to buy. The Recommendation system will recommend books to a user based on popularity, users previous ratings, ratings based on similar users. The dataset used is 7K Books. It contains a table named books which has 12 columns with some missing values in it. A recommendation system is one of the most powerful marketing tools in today's digital world. Big companies from Netflix, to Amazon use these recommendation engines to suggest products and services to their customers.

Implementation:

```
import streamlit as st
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import neighbors
from sklearn.preprocessing import MinMaxScaler
from skimage import io
import cv2
import base64


df = pd.read_csv('books.csv',error_bad_lines = False)
df = df.fillna(0)

# making copy of dataframe
df2 = df.copy()

df2.loc[ (df2['average_rating'] >= 0) & (df2['average_rating'] <= 1),
'rating_between'] = "between 0 and 1"
df2.loc[ (df2['average_rating'] > 1) & (df2['average_rating'] <= 2),
'rating_between'] = "between 1 and 2"
df2.loc[ (df2['average_rating'] > 2) & (df2['average_rating'] <= 3),
'rating_between'] = "between 2 and 3"
df2.loc[ (df2['average_rating'] > 3) & (df2['average_rating'] <= 4),
'rating_between'] = "between 3 and 4"
df2.loc[ (df2['average_rating'] > 4) & (df2['average_rating'] <= 5),
'rating_between'] = "between 4 and 5"

rating_df = pd.get_dummies(df2['rating_between'])
rating_df = rating_df.replace(np.nan, 0)

features = pd.concat([rating_df, df2['average_rating'],
                       df2['ratings_count']], axis=1)
```

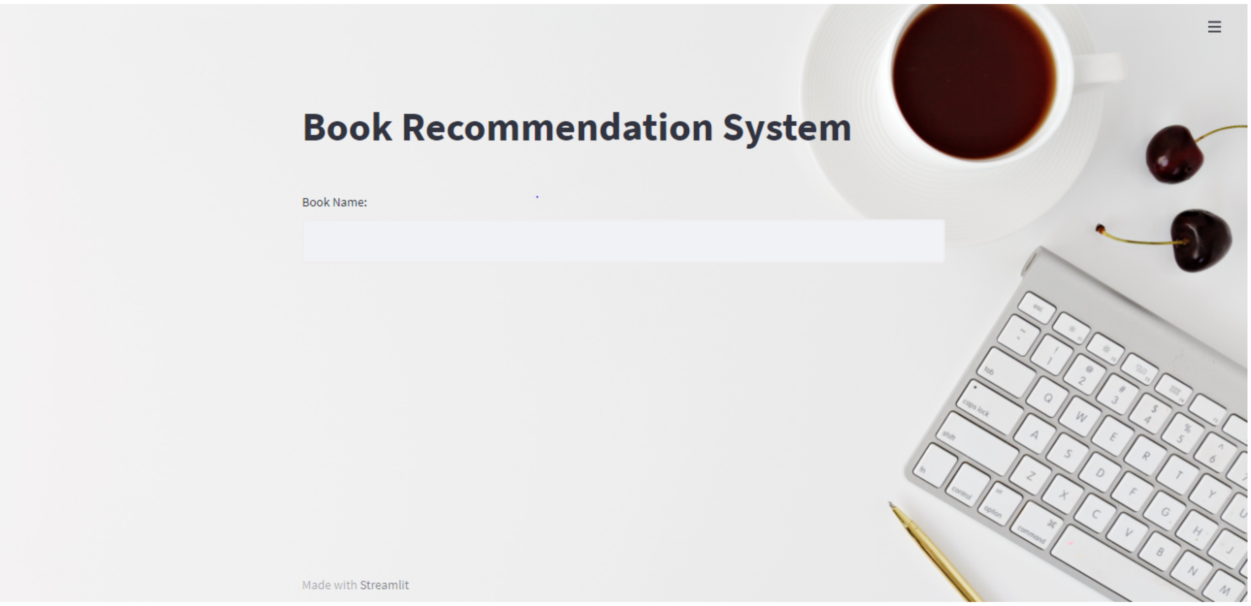


```
min_max_scaler = MinMaxScaler()
features = min_max_scaler.fit_transform(features)

# KNN model
model = neighbors.NearestNeighbors(n_neighbors=6, algorithm='auto')
model.fit(features)
dist, idlist = model.kneighbors(features)

st.title('Book Recommendation System')

def crop(img, center, width, height):
    return cv2.getRectSubPix(img, (width, height), center)

# main function
def BookRecommender(book_name):
    book_list_name = []
    dictionary = dict()

    cols = st.columns(3)

    book_id = df2[df2['title'] == book_name].index
    try:
        id = book_id[0]
    except:
        id = 0

    if (id == 0):
        st.write("")
    else:
        book_id = book_id[0]
        ex = []
        i=0
        for newid in idlist[book_id]:

            book_list_name = []
            name = df2.iloc[newid]['title']
            url = df2.iloc[newid]['thumbnail']
```



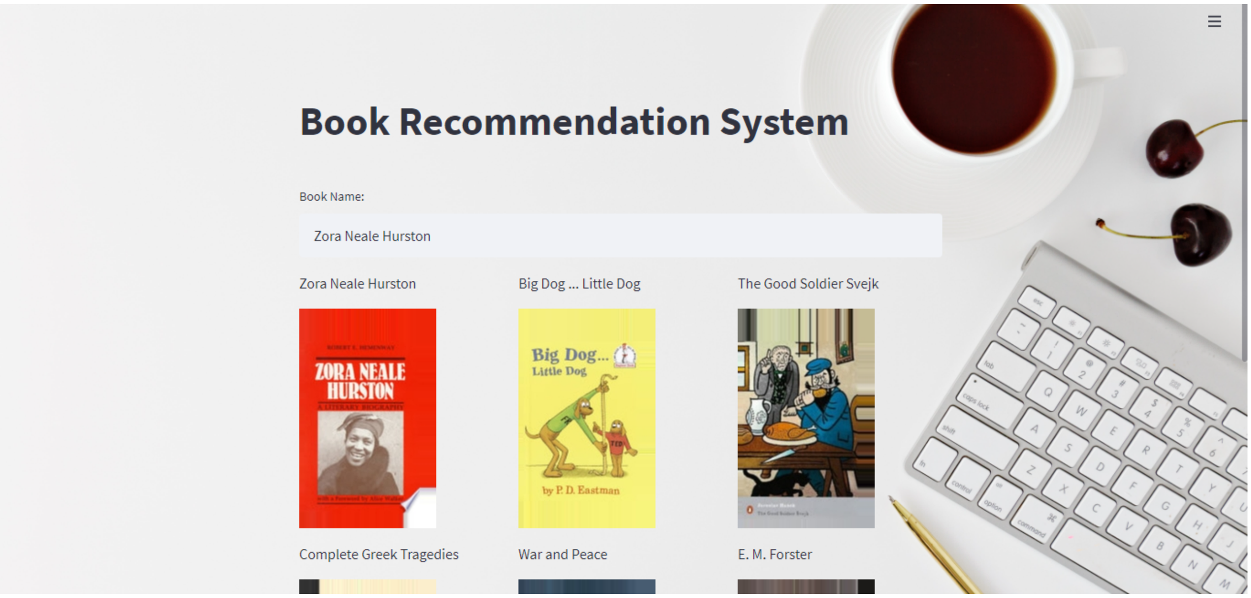
```
            cols[i].write(name)

            if ( url != 0):
                image = io.imread(url)
                img = cv2.imshow('image',image)
                cropped = crop(image,center=(70, 85), width=150,
height=240)
                cols[i].image(cropped)
                if(i==2):
                    i=0
                else:
                    i=i+1
            else:

                cols[i].image("imagenotfound.jpg",width=150)
                if(i==2):
                    i=0
                else:
                    i=i+1
# front end
st.markdown("""
<style>
body {
  color: red;
}
</style>
    """, unsafe_allow_html=True)

main_bg = "bgimg.jpg"
main_bg_ext = "jpg"

st.markdown(
    f"""
    <style>
    .reportview-container {{
        background:
url(data:image/{main_bg_ext};base64,{base64.b64encode(open(main_bg,
"rb").read()).decode()});
        background-size:cover;
    }}

    </style>
    """,
    unsafe_allow_html=True
)

book_name = st.text_input("Book Name:", "")
BookNames = BookRecommender(book_name)
```

Visualization:



Book Recommendation System

Book Name:



Book Recommendation System

Book Name:

Zora Neale Hurston

Zora Neale Hurston

Big Dog ... Little Dog

The Good Soldier Svejk

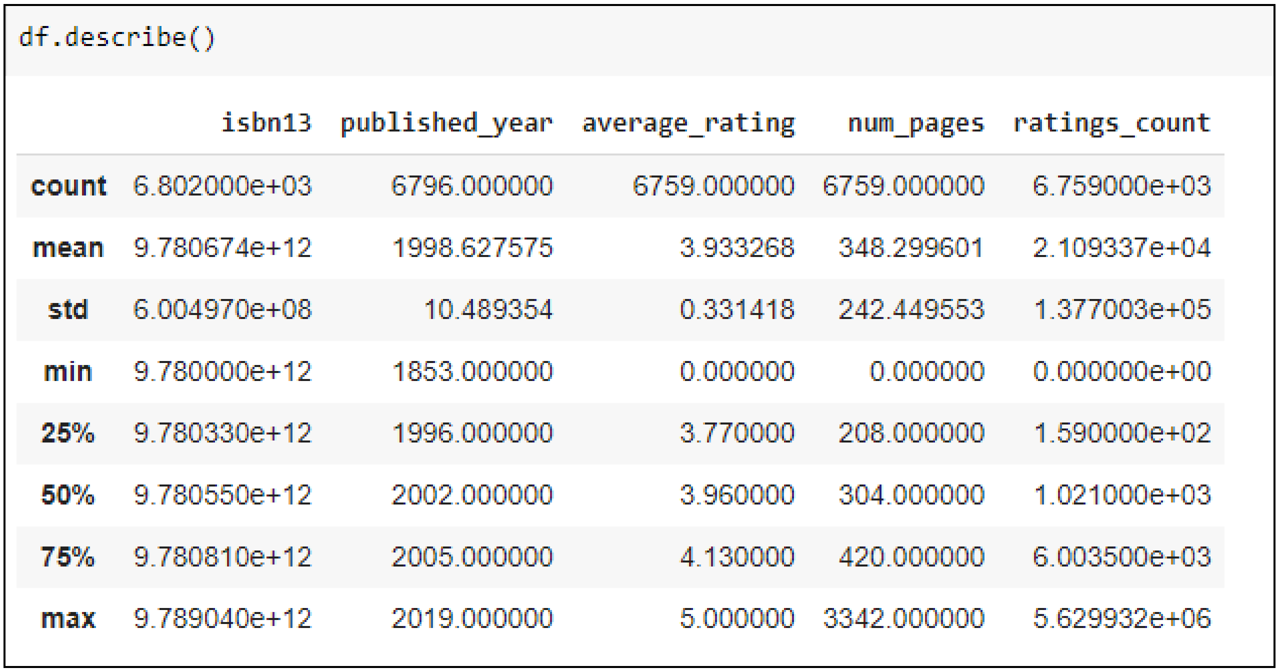
Complete Greek Tragedies

War and Peace

E. M. Forster

Results and Analysis:

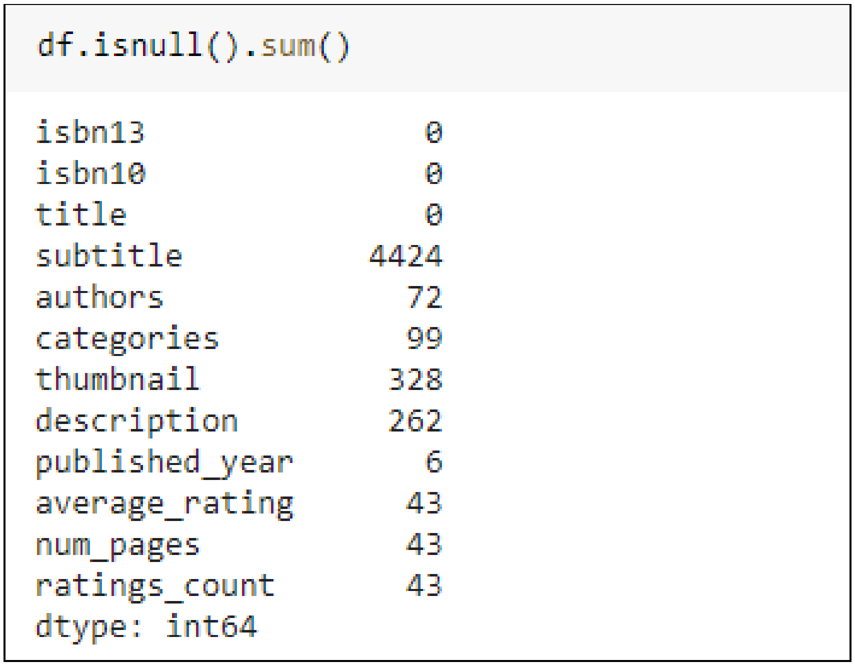
1. From the below image, we can see that the scores are all between 0 and 5. We also get to know more about the other columns, such as the average of the mean scores and other information that might help us in the next steps.



```
df.describe()
```

```
             isbn13  published_year  average_rating    num_pages  ratings_count
count  6.802000e+03     6796.000000     6759.000000  6759.000000   6.759000e+03
mean   9.780674e+12     1998.627575        3.933268   348.299601   2.109337e+04
std    6.004970e+08       10.489354        0.331418   242.449553   1.377003e+05
min    9.780000e+12     1853.000000        0.000000     0.000000   0.000000e+00
25%    9.780330e+12     1996.000000        3.770000   208.000000   1.590000e+02
50%    9.780550e+12     2002.000000        3.960000   304.000000   1.021000e+03
75%    9.780810e+12     2005.000000        4.130000   420.000000   6.003500e+03
max    9.789040e+12     2019.000000        5.000000  3342.000000   5.629932e+06
```

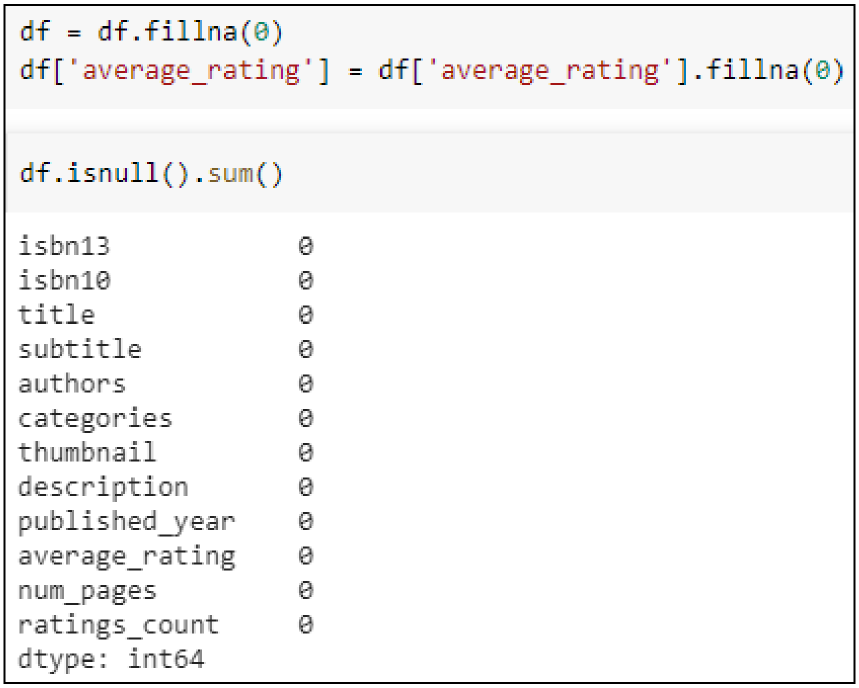
2. Check if there are any null values present in our data



```
df.isnull().sum()
```

```
isbn13              0
isbn10              0
title               0
subtitle         4424
authors            72
categories         99
thumbnail         328
description       262
published_year      6
average_rating     43
num_pages          43
ratings_count      43
dtype: int64
```

3. Eliminate Null values



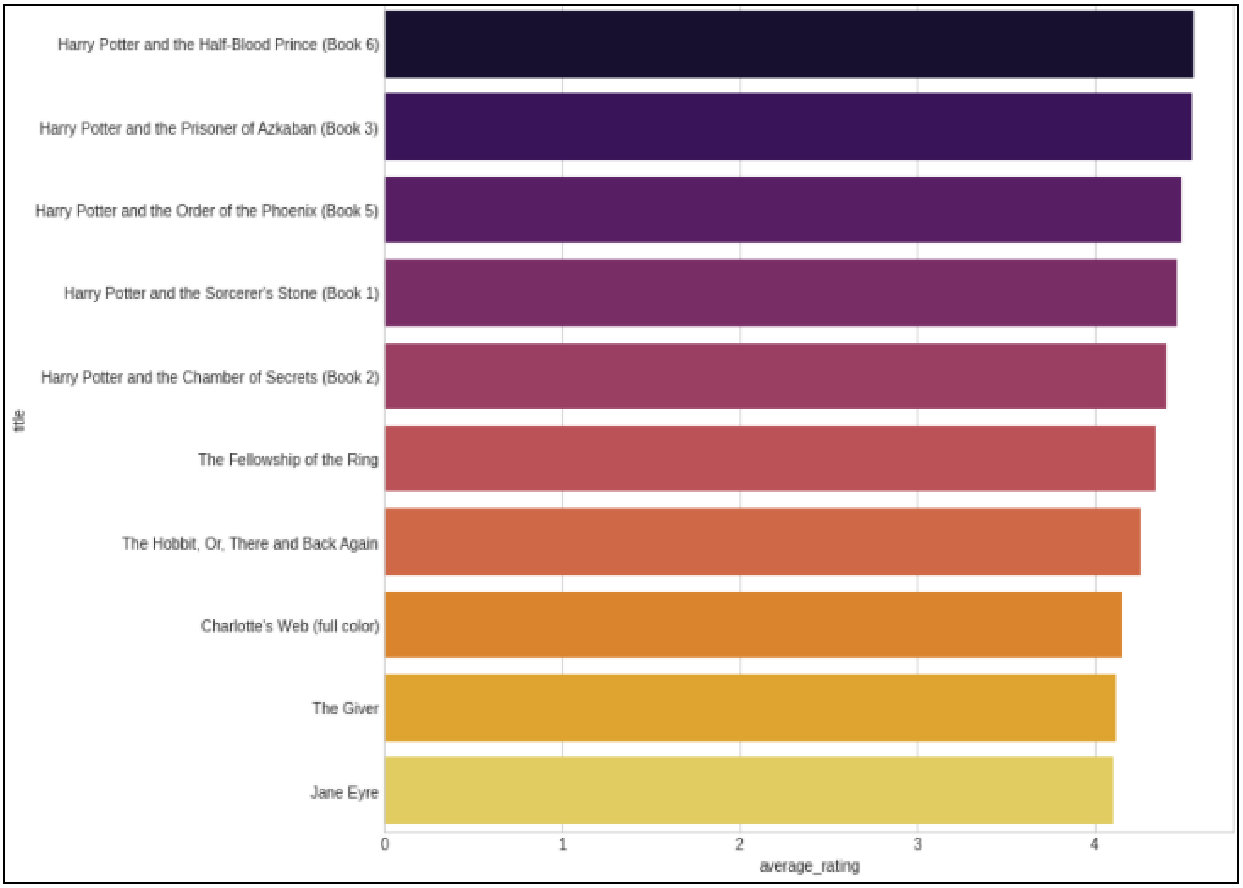
```
df = df.fillna(0)
df['average_rating'] = df['average_rating'].fillna(0)
```

```
df.isnull().sum()
```

```
isbn13            0
isbn10            0
title             0
subtitle          0
authors           0
categories        0
thumbnail         0
description       0
published_year    0
average_rating    0
num_pages         0
ratings_count     0
dtype: int64
```

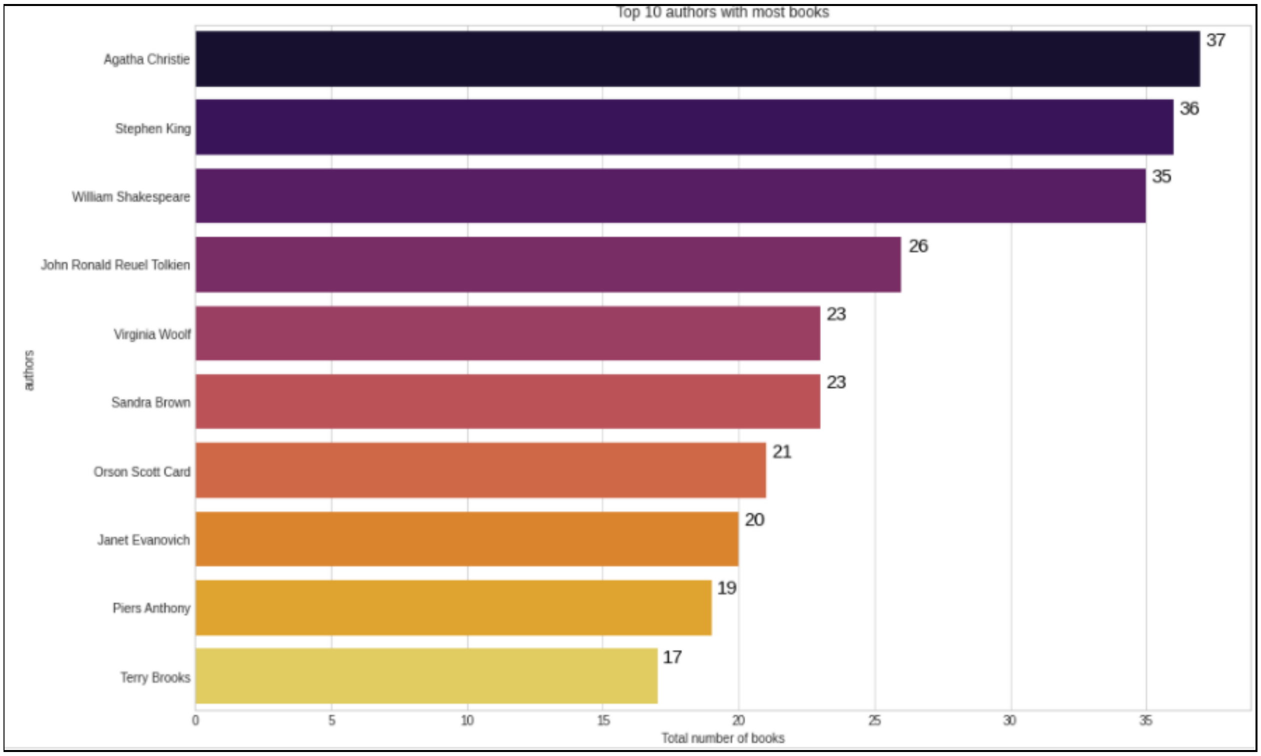
4. The results below show us the top 10 books in our data. We saw that the max score in our data was 5.0 but we don't see any books in the above result with a score of 5.0. we have filtered these books according to the number of notes. We've made sure that all of the books we have in the above results have a decent rating. There may be books in the data

that may have only 1 or 2 notes and may be rated 5.0. We want to avoid such books, which is why we used this type of filtering

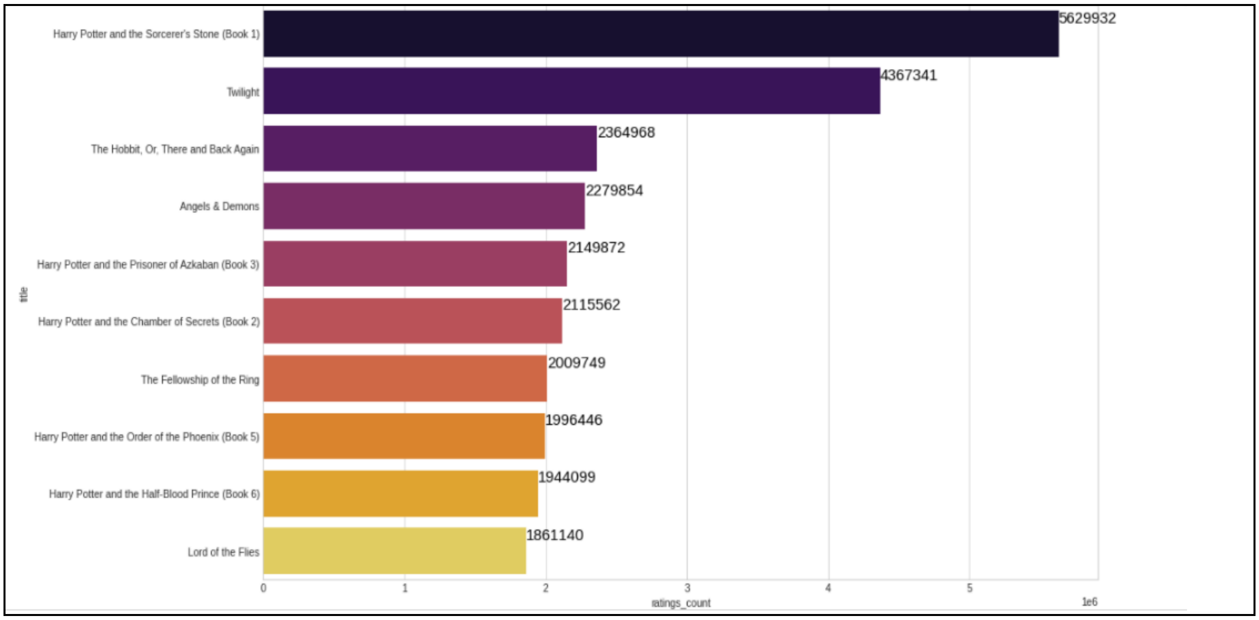


.

5. The results below show the top 10 authors in our dataset. From the chart, Agastha Christie and Stephen King have the most books in the data.

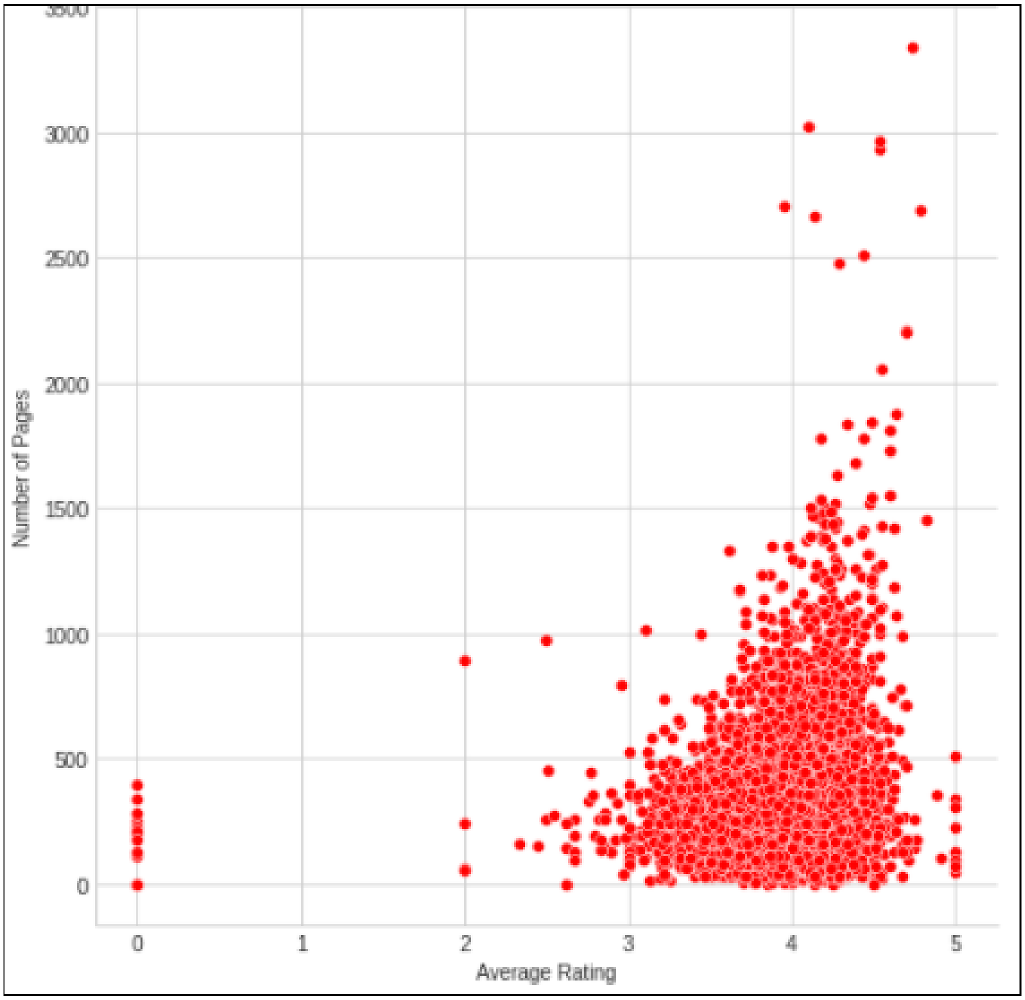


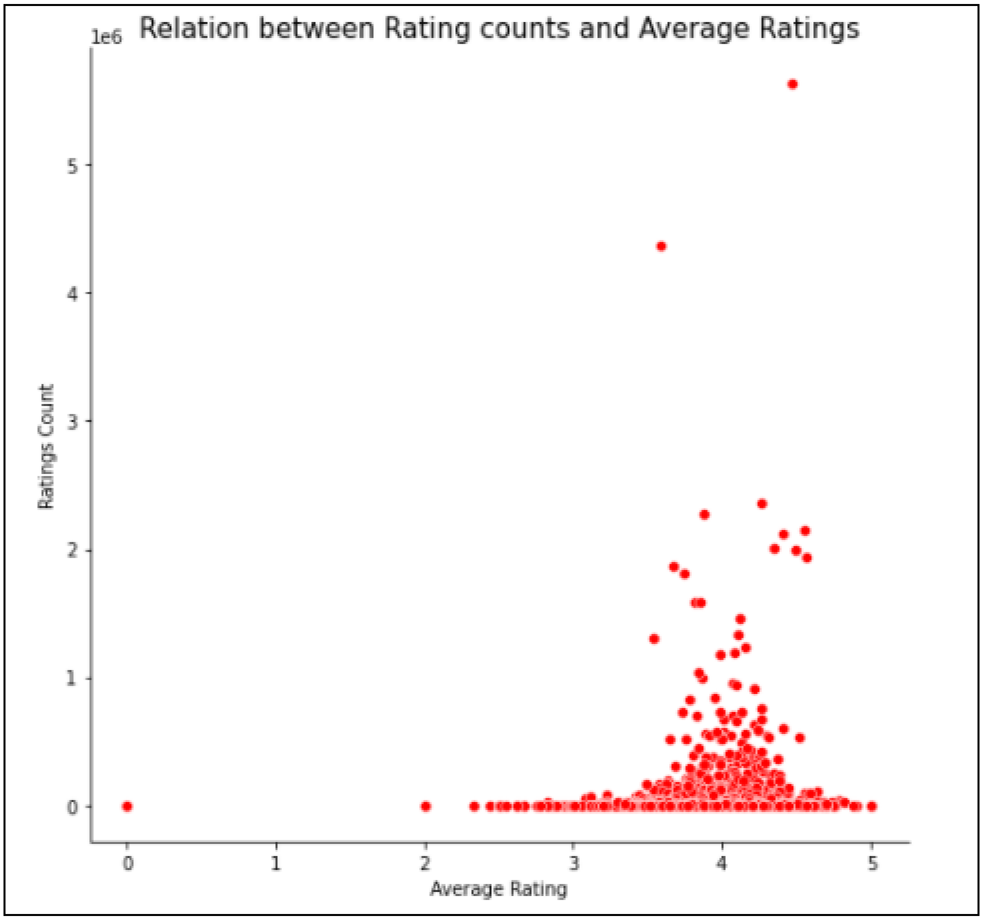
6. The results below show the top 10 books based on Ratings count.



7. After comparing the average rating with the different columns such as number of pages,ratings count, we decided to use the number of ratings for our recommendation

system. Still, the other columns didn’t make much sense and using them might not help us to a great extent so we could omit them.





Conclusion:

Recommendation systems have been important in E-commerce on the web for the customer to suggest items that they would be interested in. With the increasing number of users and items, recommendation systems encounter the main shortcoming: data sparsity and data scalability problems, which bring out the poor quality of prediction and the inefficient time consumption.

Online book reading and selling websites like Kindle and Goodreads compete against each other on many factors. One of those important factors is their book recommendation system. A book recommendation system is designed to recommend books of interest to the buyer. Thus, we built a book recommendation system with Machine Learning.

References:-

1. https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-the-fundamentals-dfee054f978e
2. https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning
3. https://www.kaggle.com/sankha1998/collaborative-book-recommendation-system
4. https://www.kaggle.com/dylanjcastillo/7k-books-with-metadata
5. https://codeburst.io/explanation-of-recommender-systems-in-information-retrieval-13077e1d916c