```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


Books = pd.read_csv('/content/BOOKS.csv')
Ratings = pd.read_csv('/content/Books_Ratings.csv')
Users = pd.read_csv('/content/Books_Users.csv')

Books.head()

Ratings.head()

Users.head()

print(Books.shape)
print(Ratings.shape)
print(Users.shape)

Books.isnull().sum()

Users.isnull().sum()

Ratings.isnull().sum()

Books.duplicated().sum()

Ratings.duplicated().sum()

Users.duplicated().sum()
```

Popularity Based Recoomender System

```
Ratings.merge(Books,on='ISBN').shape

ratings_with_name = Ratings.merge(Books,on='ISBN')

ratings_with_name.groupby('Book-Title').count()

book_ratingCount = (ratings_with_name.
     groupby(by = ['Book-Title'])['Book-Rating'].
     count().
     reset_index().
     rename(columns = {'Book-Rating': 'totalRatingCount'})
     [['Book-Title', 'totalRatingCount']]
    )
print(book_ratingCount.head())


avg_rating_df= ratings_with_name.groupby('Book-Title').mean()['Book-Rating'].reset_index()
avg_rating_df.rename(columns={'Book-Rating':'avg_ratings'},inplace=True)
avg_rating_df

popular_df= book_ratingCount.merge(avg_rating_df,on='Book-Title')
popular_df
```

```
popular_df[popular_df['totalRatingCount']>=250]

popular_df=popular_df.merge(Books,on='Book-Title')

popular_df

popular_df

popular_df.drop(columns=['ISBN','Year-Of-Publication','Image-URL-S', 'Image-URL-L'])
```

Collaberative Filtering Based Recommender System

```
x= ratings_with_name.groupby('User-ID').count()['Book-Rating']>200
x[x]

educated_users=x[x].index

ratings_with_name['User-ID'].isin(educated_users)

filtered_rating = ratings_with_name[ratings_with_name['User-ID'].isin(educated_users)]

y = filtered_rating.groupby('Book-Title').count()['Book-Rating']>=50
y[y]
```

Double-click (or enter) to edit

```
famous_books = y[y].index

filtered_rating[filtered_rating['Book-Title'].isin(famous_books)]

final_ratings= filtered_rating[filtered_rating['Book-Title'].isin(famous_books)]

final_ratings.drop_duplicates()

 PT = final_ratings.pivot_table(index='Book-Title',columns= 'User-ID',values='Book-Rating')

PT.fillna(0,inplace=True)

PT

from sklearn.metrics.pairwise import cosine_similarity

similarity_scores= cosine_similarity(PT)


similarity_scores.shape

def recommenend(book_name):
  #index fetch
   return suggestions

np.where(PT.index=='1st to Die: A Novel')
```

```
np.where(PT.index=='Wild Animus')

similarity_scores[0]

list(enumerate(similarity_scores[0]))

def recommenend(book_name):
  index = np.where(PT.index==book_name)[0][0]
  similar_items = sorted(list(enumerate(similarity_scores[index])),key=lambda x:x[1])[1:6]

  for i in similar_items:
    print(PT.index[i[0]])

recommenend('1st to Die: A Novel')

PT.index[95]

recommenend('Message in a Bottle')

recommenend('The Da Vinci Code')

import pickle
pickle.dump(popular_df,open('popular.pickle', 'wb'))
pickle.dump(popular_df,open('PT.pickle', 'wb'))
pickle.dump(popular_df,open('books.pickle', 'wb'))
pickle.dump(similarity_scores,open('similarity_scores.pkl','wb'))

from google.colab.output import eval_js
print(eval_js("google.colab.kernel.proxyPort(5000)"))

from flask import Flask,render_template,request
import pickle
import numpy as np

popular_df = pickle.load(open('/content/popular.pickle','rb'))
pt = pickle.load(open('/content/PT.pickle','rb'))
books = pickle.load(open('/content/books.pickle','rb'))
similarity_scores = pickle.load(open('/content/similarity_scores.pkl','rb'))

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html',
                           book_name = list(popular_df['Book-Title'].values),
                           author=list(popular_df['Book-Author'].values),
                           image=list(popular_df['Image-URL-M'].values),
                           votes=list(popular_df['num_ratings'].values),
                           rating=list(popular_df['avg_rating'].values)
                           )

@app.route('/recommend')
def recommend_ui():
    return render_template('recommend.html')

@app.route('/recommend_books',methods=['post'])
def recommend():
    user_input = request.form.get('user_input')
    index = np.where(pt.index == user_input)[0][0]
```

```
    similar_items = sorted(list(enumerate(similarity_scores[index])), key=lambda x: x[1], revers

    data = []
    for i in similar_items:
        item = []
        temp_df = books[books['Book-Title'] == pt.index[i[0]]]
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Title'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Author'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Image-URL-M'].values))

        data.append(item)

    print(data)

    return render_template('recommend.html',data=data)

if __name__ == '__main__':
    app.run(debug=True)
```

Colab paid products - Cancel contracts here