Project: Online Book Recommendation System

Business Objective/Problem

- Một doanh nghiệp có website bán sách đang có nhu cầu tăng doanh thu thông qua việc bán sách
  => Họ muốn xây dựng hệ thống giới thiệu sách để có thể giới thiệu các cuốn sách hữu ích có liên quan đến nhu cầu của độc giả nhằm tăng khả năng mua sách của các độc giả.

Triển khai dự án

Bước 1: Business Understanding

Dựa vào mô tả nói trên (hoặc sau khi đặt ra các câu hỏi cụ thể cho các đối tượng có liên quan) => xác định được vấn đề:

- Hiện tại: Website đang cung cấp các cuốn sách của nhiều tác giả khác nhau với nhiều loại ngôn ngữ (đa số là sách tiếng Anh).
- Độc giả của website có độ tuổi trung bình là 36 đến từ nhiều quốc gia khác nhau.
  =>**Mục tiêu/ Vấn đề:** Xây dựng mô hình giới thiệu 5 cuốn sách khác có liên quan cho độc giả khi họ đang xem một cuốn sách nào đó.

Bước 2: Data Understanding/ Acquire

Từ mục tiêu/ vấn đề đã xác định: xem xét các dữ liệu đang có:

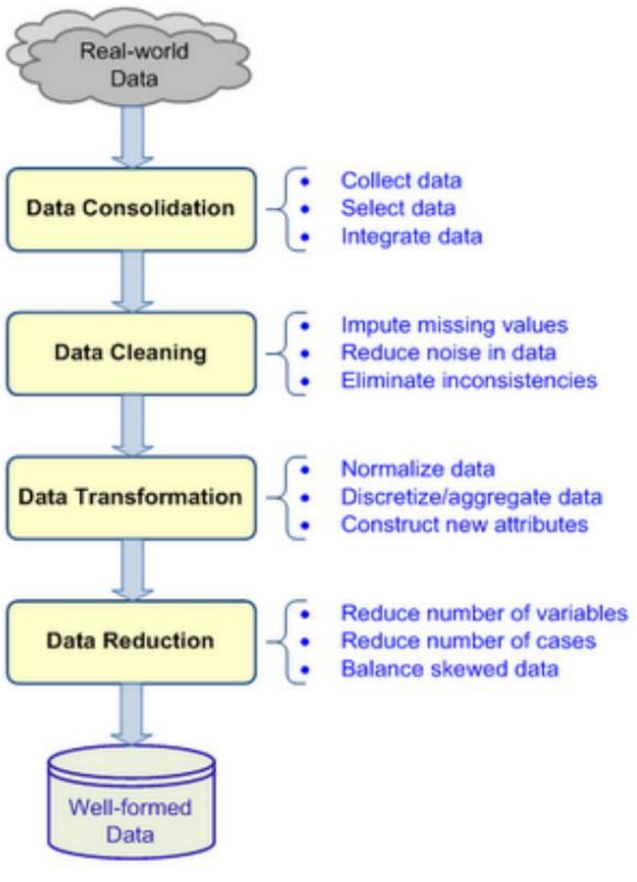
- Dữ liệu được lấy trực tiếp từ bộ phận quản lý dữ liệu của website.
- Dữ liệu đại diện cho dữ liệu đánh giá chất lượng sách (rating) của các độc giả đến từ nhiều quốc gia khác nhau với độ tuổi trung bình là 36.
- Mỗi độc giả được quản lý bởi một mã định danh duy nhất (user_id).
- Mỗi cuốn sách được quản lý bởi một mã số tiêu chuẩn quốc tế duy nhất (isbn).

Toàn bộ dữ liệu được đổ ra và lưu trữ trong tập tin books_5.csv với 51559 record. Bao gồm các cột:

- user_id — Id của độc giả
- location — thông tin vị trí của độc giả
- age — Tuổi của độc giả
- isbn — Mã số tiêu chuẩn quốc tế
- rating — Điểm đánh giá của độc giả cho cuốn sách
- book_title — Tiêu đề cuốn sách
- book_author — Tên tác giả
- year_of_publication — Năm xuất bản
- publisher — Nhà xuất bản
- img_s, img_m, img_l — Hình bìa (small, medium, large)
- Summary — Nội dung mô tả sách
- Language — Ngôn ngữ của cuốn sách
- Category — Thể loại sách
- city — Thành phố nơi độc giả đang sống
- state — bang/tỉnh nơi độc giả đang sống
- country — Quốc gia nơi độc giả đang sống

In [ ]:
```
!pip install pandas-profiling==2.7.1
```

Bước 3: Data preparation/ Prepare



In [2]:
```
#Import các thư viện cần thiết
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
import requests

import pandas_profiling as pp
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.cluster import KMeans
from sklearn.neighbors import NearestNeighbors

import re
```

In [3]:
```
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

In [4]:
```
pd.options.display.float_format = '{:.2f}'.format
```

In [5]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice_2023/Chapter16_DS_Process/data/'
```

```
Mounted at /content/gdrive
```

In [7]:
```
df = pd.read_csv('books_5.csv')
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51559 entries, 0 to 51558
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   user_id              51559 non-null  int64  
 1   location             51559 non-null  object 
 2   age                  51559 non-null  float64
 3   isbn                 51559 non-null  object 
 4   rating               51559 non-null  int64  
 5   book_title           51559 non-null  object 
 6   book_author          51559 non-null  object 
 7   year_of_publication  51559 non-null  float64
 8   publisher            51559 non-null  object 
 9   img_s                51559 non-null  object 
 10  img_m                51559 non-null  object 
 11  img_l                51559 non-null  object 
 12  Summary              51559 non-null  object 
 13  Language             51559 non-null  object 
 14  Category             51559 non-null  object 
 15  city                 50825 non-null  object 
 16  state                50385 non-null  object 
 17  country              49728 non-null  object 
dtypes: float64(2), int64(2), object(14)
memory usage: 7.1+ MB
```

In [9]:
```
df.head()
```

Out[9]:

	user_id	location	age	isbn	rating	book_title	book_author	year_of_publication	publisher	img_s
0	212898	la ronge, saskatchewan, canada	34.74	0671648187	0	STORMY IN THE WEST	Norman A. Fox	1989.00	Pocket	http://images.amazon.com/images/P/0671648187.0...	ht
1	180348	fredericksburg, virginia, usa	40.00	0505523647	0	Once A Pirate	Susan Grant	2000.00	Dorchester Publishing Company	http://images.amazon.com/images/P/0505523647.0...	ht
2	55493	chatham, ontario, canada	35.00	0138102767	6	The Silent Twins: A true story of love and hat...	Marjorie Wallace	1986.00	Simon & Schuster	http://images.amazon.com/images/P/0138102767.0...	ht
3	233398	branson, missouri, usa	24.00	0842345523	7	More Than a Carpenter	Josh McDowell	1987.00	Tyndale House Publishers	http://images.amazon.com/images/P/0842345523.0...	ht
4	248531	canby, oregon, usa	47.00	006092411X	4	The Living : A Novel	Annie Dillard	1993.00	Perennial	http://images.amazon.com/images/P/006092411X.0...	ht

First EDA => Check data

In [1]:
```
# thực hiện EDA ban đầu với: ProfileReport
# report = pp.ProfileReport(df)
# report
```

Từ kết quả của Pandas Profiling Report chúng ta có thể thấy:

- Dữ liệu khá đầy đủ thông tin, Chỉ có các cột city, state và country có missing value với tỉ lệ lần lượt là 1.4%, 2.3% và 3.6%
- Không có dữ liệu trùng
- Chỉ có 4 biến Continuous là: user_id, age, rating và year_of_publication

In [11]:
```
#kiểm tra lại trên DataFrame
df.isnull().any()
```

Out[11]:
```
user_id              False
location             False
age                  False
isbn                 False
rating               False
book_title           False
book_author          False
year_of_publication  False
publisher            False
img_s                False
img_m                False
img_l                False
Summary              False
Language             False
Category             False
city                  True
state                 True
country               True
dtype: bool
```

In [12]:
```
df.isna().any()
```

Out[12]:
```
user_id              False
location             False
age                  False
isbn                 False
rating               False
book_title           False
book_author          False
year_of_publication  False
publisher            False
img_s                False
img_m                False
img_l                False
Summary              False
Language             False
Category             False
city                  True
state                 True
country               True
dtype: bool
```

In [13]:
```
#kiểm tra tỉ lệ null
df.isnull().sum()/df.shape[0] * 100
```

Out[13]:
```
user_id              0.00
location             0.00
age                  0.00
isbn                 0.00
rating               0.00
book_title           0.00
book_author          0.00
year_of_publication  0.00
publisher            0.00
img_s                0.00
img_m                0.00
img_l                0.00
Summary              0.00
Language             0.00
Category             0.00
city                 1.42
state                2.28
country              3.55
dtype: float64
```

In [14]:
```
df.duplicated().any()
```

Out[14]:
```
False
```

In [15]:
```
df.describe(include='all')
```

Out[15]:

	user_id	location	age	isbn	rating	book_title	book_author	year_of_publication	publisher	im
count	51559.00	51559	51559.00	51559	51559.00	51559	51559	51559.00	51559	51
unique	NaN	6766	NaN	35182	NaN	32847	17501	NaN	3879	35
top	NaN	toronto, ontario, canada	NaN	0971880107	NaN	Wild Animus	Stephen King	NaN	Ballantine Books	http://images.amazon.com/images/P/0971880107
freq	NaN	765	NaN	114	NaN	114	523	NaN	1703	
mean	140611.19	NaN	36.32	NaN	2.82	NaN	NaN	1995.21	NaN	I
std	80330.17	NaN	10.28	NaN	3.85	NaN	NaN	7.36	NaN	I
min	8.00	NaN	5.00	NaN	0.00	NaN	NaN	1900.00	NaN	I
25%	71712.00	NaN	31.00	NaN	0.00	NaN	NaN	1992.00	NaN	I
50%	140000.00	NaN	34.74	NaN	0.00	NaN	NaN	1997.00	NaN	I
75%	211847.00	NaN	41.00	NaN	7.00	NaN	NaN	2001.00	NaN	I
max	278851.00	NaN	97.00	NaN	10.00	NaN	NaN	2005.00	NaN	I

Loại bỏ các cột không cần thiết

In [16]:
```
#Loại bỏ các cột không có nhiều ý nghĩa trong việc đề xuất sách cho người dùng
df.drop(columns = ['location','isbn', 'img_s','img_l','city','age', 'state','Language','country', 'year_of_publication'], inplace
```

In [17]:
```
#Loại bỏ các ký tự không cần thiết trong Category
# df['Category'] = df['Category'].apply(lambda x: re.sub('[\W_]+',' ',x).strip())
df['Category'] = df['Category'].str.replace('[\W_]+',' ').str.strip()
```

In [18]:
```
df['Category'].value_counts()
```

Out[18]:
```
9                          20369
Fiction                    19070
Juvenile Fiction            2013
Biography Autobiography     1153
History                      438
                           ...
Children of clergy             1
Bath time                      1
Actors and actresses           1
Call of Cthulhu Game           1
Giants                         1
Name: Category, Length: 1408, dtype: int64
```

Khám phá các từ quan trọng trong book_title

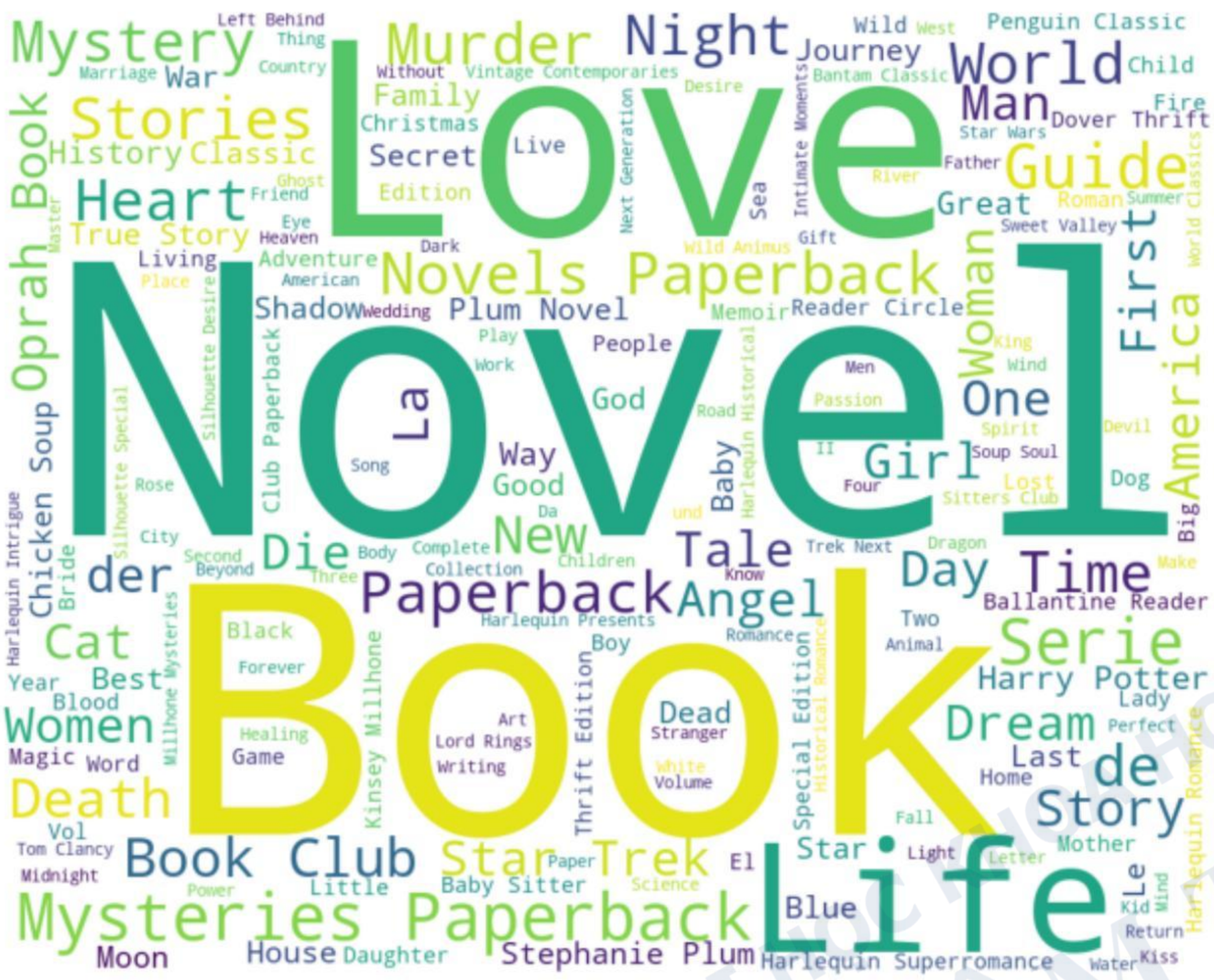
In [19]:
```
text = " ".join(df['book_title'].to_list())

# Create stopword list
stopword_list = set(list(STOPWORDS) + ['br',])

# Create WordCloud
word_cloud = WordCloud(width = 1000, height = 800, background_color ='White',
                       stopwords = stopword_list, min_font_size = 14)

word_cloud.generate(text)

plt.figure(figsize = (15, 9))
plt.imshow(word_cloud)
plt.axis("off")
plt.show()
```



Khám phá các từ quan trọng trong Summary

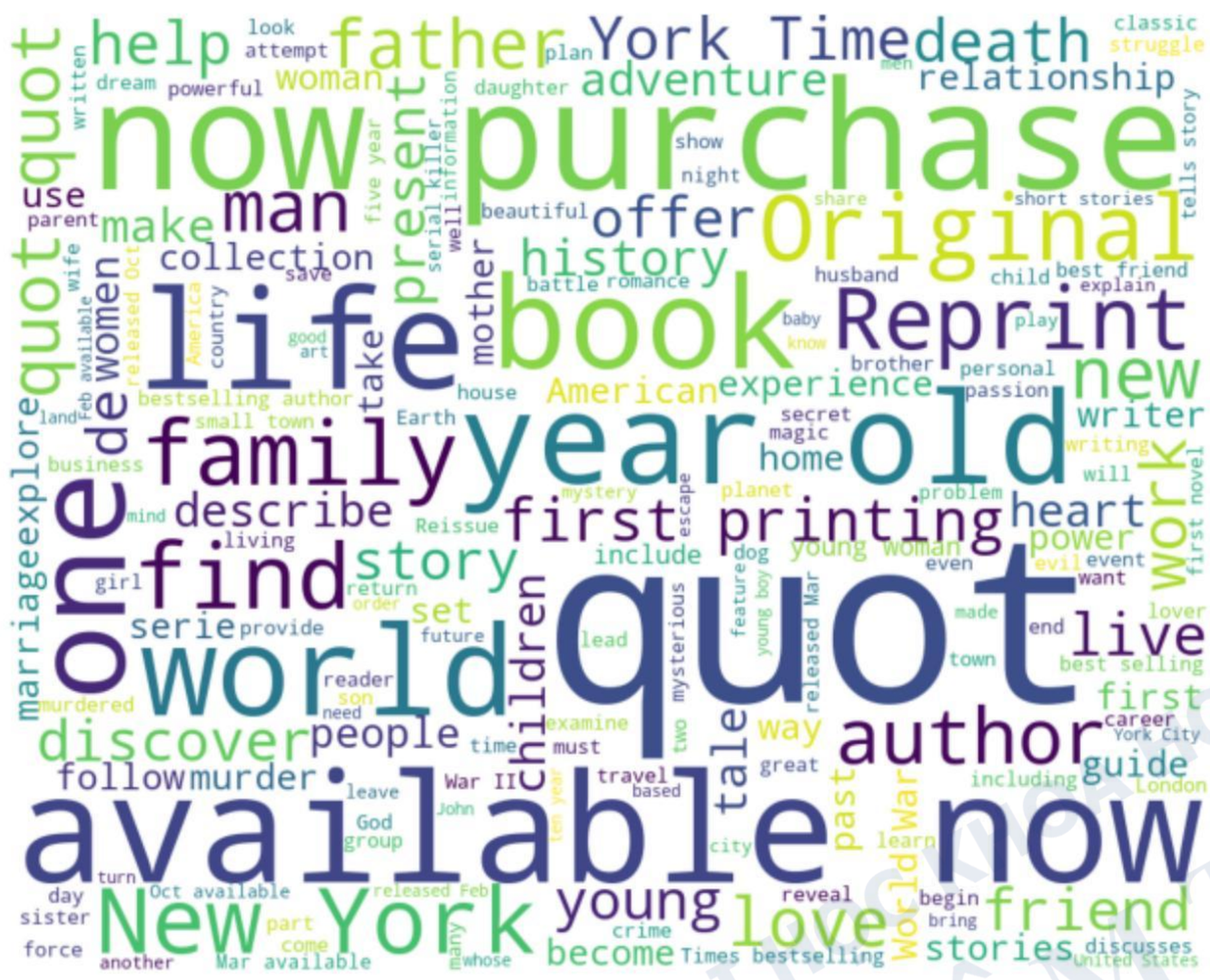
In [20]:
```
text = " ".join(df['Summary'].to_list())

# Create stopword list
stopword_list = set(list(STOPWORDS) + ['br',])

# Create WordCloud
word_cloud = WordCloud(width = 1000, height = 800, background_color ='White',
                       stopwords = stopword_list, min_font_size = 14)

word_cloud.generate(text)

plt.figure(figsize = (15, 9))
plt.imshow(word_cloud)
plt.axis("off")
plt.show()
```



Feature Engineering

In [21]:
```
data = df[['book_title', 'book_author', 'publisher',
           'img_m', 'Summary', 'Category']].copy(deep=True)
```

In [22]:
```
data.duplicated(subset=['book_title', 'book_author',
                        'Summary', 'Category']).sum()
```

Out[22]:
```
16765
```

In [23]:
```
data.drop_duplicates(subset=['book_title', 'book_author', 'Summary', 'Category'],
                                    keep='first', inplace=True)
data.reset_index(drop=True, inplace=True)
```

In [24]:
```
# Tạo cột book_title + Summary
data['title_summary'] = data['book_title'] + ' ' + data['Summary']
```

In [25]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34794 entries, 0 to 34793
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   book_title     34794 non-null  object
 1   book_author    34794 non-null  object
 2   publisher      34794 non-null  object
 3   img_m          34794 non-null  object
 4   Summary        34794 non-null  object
 5   Category       34794 non-null  object
 6   title_summary  34794 non-null  object
dtypes: object(7)
memory usage: 1.9+ MB
```

In [26]:
```
%%time
# Biến đổi cột title_summary sử dụng TF_IDF, trả về 1000 từ quan trọng nhất
tfidf = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}',
                        ngram_range=(1, 3), stop_words = 'english',
                        max_features=1000)

tfidf_matrix = tfidf.fit_transform(data['title_summary'])
tfidf_matrix.shape
```

```
CPU times: user 5.68 s, sys: 160 ms, total: 5.84 s
Wall time: 5.84 s
```

Bước 4&5: Modeling & Evaluation/ Analyze & Report

In [27]:
```
k = 5 # Số sách đề xuất
model = NearestNeighbors(n_neighbors=k)
```

In [28]:
```
%%time
model.fit(tfidf_matrix)
dist, idlist = model.kneighbors(tfidf_matrix)
```

```
CPU times: user 23.8 s, sys: 1.01 s, total: 24.8 s
Wall time: 24.8 s
```

In [29]:
```
#Hàm đề xuất sách
def BookRecommender(book_name):
    book_list_name = []
    book_list_id = []
    book_id = data[data['book_title'].str.lower() == book_name.lower()].index
    book_id = book_id[0]
    for newid in idlist[book_id]:
        book_list_name.append(data.loc[newid, 'book_title'])
        book_list_id.append(newid)
    return book_list_id, book_list_name
```

In [30]:
```
book_id = data[data['book_title'].str.lower() == "Life of Pi".lower()].index
book_id
```

Out[30]:
```
Int64Index([20, 628], dtype='int64')
```

In [31]:
```
idlist[20]
```

Out[31]:
```
array([31295, 16616, 28940, 14430,  6630])
```

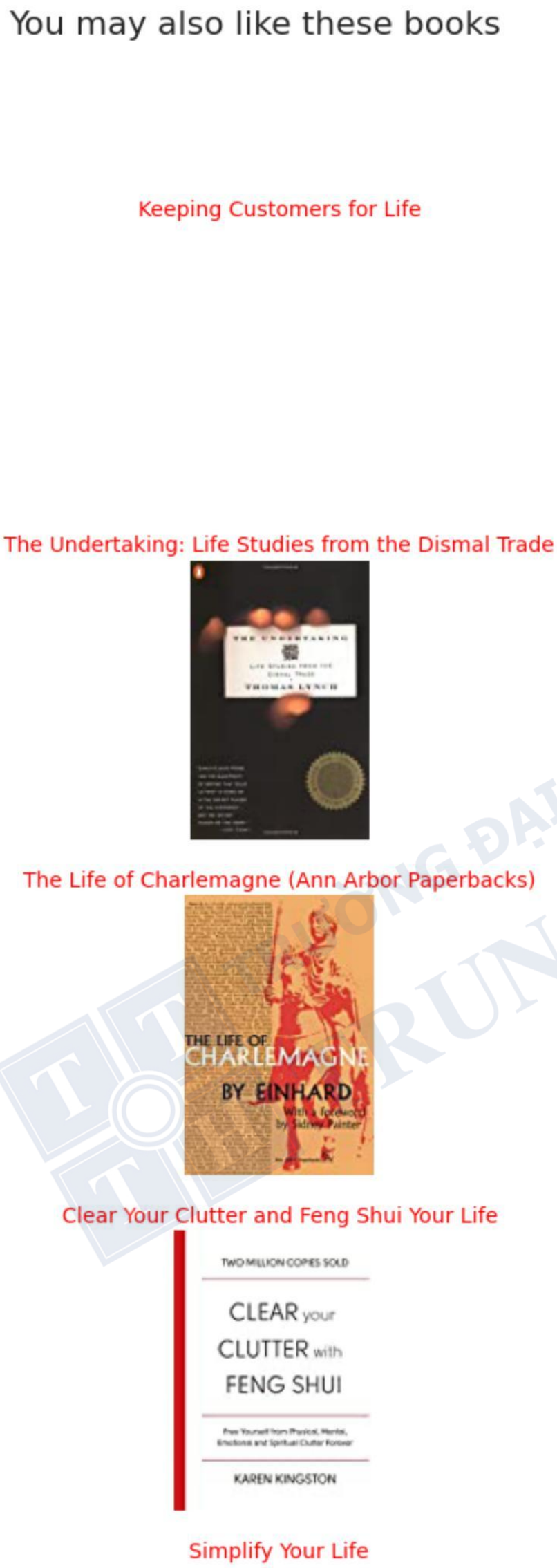
In [32]:
```
data.loc[16616, 'book_title']
```

Out[32]:
```
'The Undertaking: Life Studies from the Dismal Trade'
```

In [33]:
```
BookIds, BookNames = BookRecommender("Life of Pi")
BookNames
```

Out[33]:
```
['Keeping Customers for Life',
 'The Undertaking: Life Studies from the Dismal Trade',
 'The Life of Charlemagne (Ann Arbor Paperbacks)',
 'Clear Your Clutter and Feng Shui Your Life',
 'Simplify Your Life']
```

In [34]:
```
fig, axs = plt.subplots(5,figsize=(18,20))
fig.suptitle('You may also like these books', size = 22)
for i in range(len(BookIds)):
    idx = BookIds[i]
    url = data.loc[idx, 'img_m']
    im = Image.open(requests.get(url, stream=True).raw)
    axs[i].imshow(im)
    axs[i].axis("off")
    axs[i].set_title('{}'.format(data.loc[idx, 'book_title']),
                     color="red",
                     fontsize=14)
plt.show()
```



Bước 6: Deployment & Feedback/ Act