

✧ Book Recommendation System

- Content-Based Collaborative Filtering using Title, Author, Publisher, Category as features

About Dataset

Terdapat 278858 user memberikan 1149780 penilaian (explicit/implicit) terhadap 271379 buku

- user_id - id dari pengguna
- location - lokasi/alamat pengguna
- age - umur pengguna
- isbn - kode ISBN (International Standard Book Number) buku
- rating - rating dari buku
- book_title - judul buku
- book_author - penulis buku
- year_of_publication - tahun terbit buku
- publisher - penerbit buku
- img_s - gambar sampul buku (small)
- img_m - gambar sampul buku (medium)
- img_l - gambar sampul buku (large)
- Summary - ringkasan/sinopsis buku
- Language - bahasa yang digunakan buku
- Category - kategori buku
- city - kota pengguna
- state - negara bagian pengguna
- country - negara pengguna

✧ Libraries

```
%pip install opendatasets
```

```
Collecting opendatasets
  Downloading opendatasets-0.1.22-py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from opendatasets) (4.67.1)
Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packages (from opendatasets) (1.7.4.5)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from opendatasets) (8.3.1)
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (6.3.0)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2024.7.4)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (3.3.2)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (3.11)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (5.29.5)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2.9.0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2.32.4)
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (75.1.0)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (1.17.0)
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (1.3)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (2.5.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from kaggle->opendatasets) (0.5.1)
Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
Installing collected packages: opendatasets
Successfully installed opendatasets-0.1.22
```

```
import os
import re
import nltk
import requests
import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import opendatasets as od

from nltk.corpus import stopwords
nltk.download("stopwords")

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
from PIL import Image
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...  
[nltk_data] Unzipping corpora/stopwords.zip.
```

✓ Load and Check Dataset

```
od.download('https://www.kaggle.com/datasets/ruchi798/bookcrossing-dataset')
```

Please provide your Kaggle credentials to download this dataset. Learn more: <http://bit.ly/kaggle-creds>
Your Kaggle username: akuisal@gmail.com
Your Kaggle Key:
Dataset URL: <https://www.kaggle.com/datasets/ruchi798/bookcrossing-dataset>
Downloading bookcrossing-dataset.zip to ./bookcrossing-dataset
100%|██████████| 76.1M/76.1M [00:00<00:00, 1.15GB/s]

```
books = pd.read_csv('/content/bookcrossing-dataset/Books Data with Category Language and Summary/Preprocessed_data.csv')  
books.head(2)
```

	Unnamed: 0	user_id	location	age	isbn	rating	book_title	book_author	year_of_publication	publisher	
0	0	2	stockton, california, usa	18.0000	0195153448	0	Classical Mythology	Mark P. O. Morford	2002.0	Oxford University Press	http
1	1	8	timmins, ontario, canada	34.7439	0002005018	5	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	http

```
books.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1031175 entries, 0 to 1031174  
Data columns (total 19 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   Unnamed: 0            1031175 non-null int64  
1   user_id               1031175 non-null int64  
2   location              1031175 non-null object  
3   age                  1031175 non-null float64  
4   isbn                 1031175 non-null object  
5   rating               1031175 non-null int64  
6   book_title           1031175 non-null object  
7   book_author          1031175 non-null object  
8   year_of_publication  1031175 non-null float64  
9   publisher            1031175 non-null object  
10  img_s                1031175 non-null object  
11  img_m                1031175 non-null object  
12  img_l                1031175 non-null object  
13  Summary              1031175 non-null object  
14  Language              1031175 non-null object  
15  Category              1031175 non-null object  
16  city                  1017072 non-null object  
17  state                 1008377 non-null object  
18  country               995801 non-null object  
dtypes: float64(2), int64(3), object(14)  
memory usage: 149.5+ MB
```

```
print(sorted(books.rating.unique()))  
print()  
print(books.rating.value_counts())
```

```
[np.int64(0), np.int64(1), np.int64(2), np.int64(3), np.int64(4), np.int64(5), np.int64(6), np.int64(7), np.int64(8), np.int64(9), np.int64(10), np.int64(11), np.int64(12), np.int64(13), np.int64(14), np.int64(15), np.int64(16), np.int64(17), np.int64(18)]  
  
rating  
0    647323  
8     91806  
10    71227  
7     66404  
9     60780  
5     45355  
6     31689  
4      7617  
3      5118
```

```
2      2375
1      1481
Name: count, dtype: int64
```

```
books.isnull().sum()
```

	0
Unnamed: 0	0
user_id	0
location	0
age	0
isbn	0
rating	0
book_title	0
book_author	1
year_of_publication	0
publisher	0
img_s	0
img_m	0
img_l	0
Summary	0
Language	0
Category	0
city	14103
state	22798
country	35374

```
dtype: int64
```

```
print(books.Category.unique())
print()
print(books.Category.value_counts().index)
```

```
["['Social Science']" '['Actresses']" '['1940-1949']" ...
 '['Microsoft Windows NT.']" '['Merchants']" '['Alternative histories']"]

Index(['9', '['Fiction']', '['Juvenile Fiction']',
 '['Biography & Autobiography']', '['Humor']', '['History']',
 '['Religion']', '['Juvenile Nonfiction']', '['Social Science']',
 '['Body, Mind & Spirit']',
 ...,
 '['Germany (West)']', '['Abnormal reflexes']', '['Color.']),
 '['Creoles']', '['Yugoslav War, 1991-1995.']), '['Nouvelle']',
 '['Guenevere, Queen (Legendary character)']', '['Chess sets']',
 '['Muffin, Charlie (Fictitious character)']', '['Cookery (Tea)']'],
dtype='object', name='Category', length=6448)
```

```
# Title, Author, Publisher, Category as features
books.publisher.value_counts()
```

	count
publisher	
Ballantine Books	34724
Pocket	31989
Berkley Publishing Group	28614
Warner Books	25506
Harlequin	25029
...	...
Simmons-Boardman Books, Incorporated	1
Sight & Sound International	1
Outdoor Life	1
John Curley & Assoc	1
Raintree Childrens Books	1

16729 rows × 1 columns

dtype: int64

Preprocessing

```
df = books.copy()
df.dropna(inplace=True, how='any', axis=0)
df.reset_index(drop=True, inplace=True)
df.drop(columns = ['Unnamed: 0', 'location', 'isbn',
                  'img_s', 'img_m', 'img_l', 'city', 'age',
                  'state', 'Language', 'country',
                  'year_of_publication', 'Summary'], axis=1, inplace = True) #kolom yang didrop tidak akan dipakai
df.drop(index=df[df.Category == '9'].index, inplace=True)
df.drop(index=df[df.rating == 0].index, inplace=True)
df.Category = df.Category.apply(lambda x: re.sub('[\W_]+', ' ', x).strip())
df.head()
```

```
<>:10: SyntaxWarning: invalid escape sequence '\W'
<>:10: SyntaxWarning: invalid escape sequence '\W'
/tmp/ipython-input-918539235.py:10: SyntaxWarning: invalid escape sequence '\W'
  df.Category = df.Category.apply(lambda x: re.sub('[\W_]+', ' ', x).strip())
```

	user_id	rating	book_title	book_author	publisher	Category
1	8	5	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	Actresses
4	67544	8	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	Actresses
7	123629	9	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	Actresses
9	200273	8	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	Actresses
10	210926	9	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	Actresses

df.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 217314 entries, 1 to 982276
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   user_id      217314 non-null  int64
1   rating       217314 non-null  int64
2   book_title   217314 non-null  object
3   book_author  217314 non-null  object
4   publisher    217314 non-null  object
5   Category     217314 non-null  object
dtypes: int64(2), object(4)
memory usage: 11.6+ MB
```

df.isnull().sum()

	0
user_id	0
rating	0
book_title	0

```
# df.Category.value_counts()
i = 1
for idx, name in enumerate(df['Category'].value_counts().index.tolist()):
    if(i==25): break
    print(i)
    print('Name :', name)
    print('Counts :', df['Category'].value_counts()[idx])
    print('---'*8)
    i+=1
```

```
Name : True Crime
Counts : 965
```

```
-----
```

```
18
Name : Psychology
Counts : 949
```

```
-----
```

```
19
Name : Science
Counts : 933
```

```
-----
```

```
20
Name : Computers
Counts : 894
```

```
-----
```

```
21
Name : Literary Criticism
Counts : 877
```

```
-----
```

```
22
Name : Drama
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
Counts : 839
```

```
-----
```

```
23
Name : Political Science
Counts : 814
```

```
-----
```

```
24
Name : Philosophy
Counts : 772
```

```
-----
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
/tmp/ipython-input-3783056380.py:7: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future
print('Counts :', df['Category'].value_counts()[idx])
```

```
cat_list = df.Category.value_counts().index.tolist()
print(cat_list[5:20])
```

```
['Religion', 'Body Mind Spirit', 'Juvenile Nonfiction', 'Social Science', 'Business Economics', 'Family Relationships', 'Sel
```

```
df_fil = df[df.Category.isin(cat_list[5:20])]
df_fil.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 22576 entries, 694 to 982240
```

```
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   user_id      22576 non-null   int64
1   rating        22576 non-null   int64
2   book_title    22576 non-null   object
3   book_author   22576 non-null   object
4   publisher     22576 non-null   object
5   Category      22576 non-null   object
dtypes: int64(2), object(4)
memory usage: 1.2+ MB
```

```
df_fil.Category.nunique()
```



15

```
prep = df_fil.copy()
prep.sort_values('book_title')
```

	user_id	rating	book_title	book_author	publisher	Category	
958962	237883	9	Microsoft Application Architecture For Micros...	Microsoft Corporation Staff	Microsoft Press	Computers	
862827	131193	8	\$30 Film School	Michael W. Dean	Muska & Lipman Publishing	Computers	
614402	31826	10	1,000 Makers of the Millennium: The Men and Wo...	Dorling Kindersley Publishing	Dorling Kindersley	Juvenile Nonfiction	
518889	115161	10	1,000 Places to See Before You Die	Patricia Schultz	Workman Publishing	Travel	
518893	149153	7	1,000 Places to See Before You Die	Patricia Schultz	Workman Publishing	Travel	
...	
944736	216795	8	how to stop time : heroin from A to Z	Ann Marlowe	Basic Books	Psychology	
788607	87141	8	sed & awk (2nd Edition)	Dale Dougherty	O'Reilly	Computers	
390374	240054	9	street bible, the	Robert Lacey	Zondervan Publishing Company	Religion	
672661	217211	6	teach yourself...C++	Al Stevens	John Wiley & Sons Inc	Computers	

```
prep = prep.drop_duplicates('book_title')
prep.info()
print()
prep.head(4)
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13048 entries, 694 to 982240
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   user_id      13048 non-null   int64
1   rating        13048 non-null   int64
2   book_title    13048 non-null   object
3   book_author   13048 non-null   object
4   publisher     13048 non-null   object
5   Category      13048 non-null   object
dtypes: int64(2), object(4)
memory usage: 713.6+ KB
```

	user_id	rating	book_title	book_author	publisher	Category	
694	6366	7	New Vegetarian: Bold and Beautiful Recipes for...	Celia Brooks Brown	Ryland Peters & Small Ltd	Cooking	
4680	157475	8	The Therapeutic Touch: How to Use Your Hands t...	Dolores Krieger	Fireside	Health Fitness	
6002	64010	7	The Dragons of Eden: Speculations on the	Carl Sagan	Ballantine Books	Science	

Next steps: [Generate code with prep](#) [New interactive sheet](#)

```
prep['Category'] = prep['Category'].str.replace(' ', '_')
prep.head(10)
```

	user_id	rating	book_title	book_author	publisher		Category
694	6366	7	New Vegetarian: Bold and Beautiful Recipes for...	Celia Brooks Brown	Ryland Peters & Small Ltd		Cooking
4680	157475	8	The Therapeutic Touch: How to Use Your Hands t...	Dolores Krieger	Fireside		Health_Fitness
6002	64010	7	The Dragons of Eden: Speculations on the Evolu...	Carl Sagan	Ballantine Books		Science
8561	99	9	McDonald's: Behind the Arches	John F. Love	Bantam	Business_Economics	
8565	99	10	Creating Wealth : Retire in Ten Years Using AI...	Robert G. Allen	Fireside	Business_Economics	

```
book_title = prep['book_title'].tolist()
book_cat = prep['Category'].tolist()
book_pub = prep['publisher'].tolist()
book_author = prep['book_author'].tolist()
```

```
print(len(book_title))
print(len(book_cat))
print(len(book_pub))
print(len(book_author))
```

13048
13048
13048
13048

```
book_new = pd.DataFrame({
    'title': book_title,
    'author': book_author,
    'category': book_cat,
    'publisher': book_pub
})
book_new
```

	title	author	category	publisher	
0	New Vegetarian: Bold and Beautiful Recipes for...	Celia Brooks Brown	Cooking	Ryland Peters & Small Ltd	
1	The Therapeutic Touch: How to Use Your Hands t...	Dolores Krieger	Health_Fitness	Fireside	
2	The Dragons of Eden: Speculations on the Evolu...	Carl Sagan	Science	Ballantine Books	
3	McDonald's: Behind the Arches	John F. Love	Business_Economics	Bantam	
4	Creating Wealth : Retire in Ten Years Using AI...	Robert G. Allen	Business_Economics	Fireside	
...	
13043	Remote Perceptions: Out-Of-Body Experiences, R...	Angela Thompson Smith	Body_Mind_Spirit	Hampton Roads Publishing Co.	
13044	Who Speaks for Wolf: A Native American Learnin...	Paula Underwood	Social_Science	Tribe of Two Pr	
13045	On Becoming Childwise	Gary Ezzo	Family_Relationships	Multnomah	
13046	Frommer's 2000 Bahamas (Frommer's Bahamas, 2000)	Arthur Frommer	Travel	Hungry Minds, Inc	
13047	Inner Hunger: A Young Woman's Struggle Through...	Marianne Apostolides	Self_Help	W. W. Norton & Company	

Next steps:

[Generate code with book_new](#)

[New interactive sheet](#)

TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
#Init Tfidf
tf = TfidfVectorizer()
```

```
# Melakukan perhitungan idf pada data category
tf.fit(book_new['category'])

# Mapping array dari fitur index integer ke fitur nama
tf.get_feature_names_out()
```

```
array(['body_mind_spirit', 'business_economics', 'computers', 'cooking',
      'family_relationships', 'health_fitness', 'juvenile_nonfiction',
      'poetry', 'psychology', 'religion', 'science', 'self_help',
      'social_science', 'travel', 'true_crime'], dtype=object)
```

```
# Melakukan fit lalu ditransformasikan ke bentuk matrix
tfidf_matrix = tf.fit_transform(book_new['category'])
```

```
# Melihat ukuran matrix tfidf
tfidf_matrix.shape
```

```
(13048, 15)
```

```
# Mengubah vektor tf-idf dalam bentuk matriks dengan fungsi todense()
tfidf_matrix.todense()
```

```
matrix([[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., 1., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

```
# Membuat dataframe untuk melihat tf-idf matrix
# Kolom diisi dengan category book
# Baris diisi dengan nama book
```

```
pd.DataFrame(
    tfidf_matrix.todense(),
    columns=tf.get_feature_names_out(),
    index=book_new.title
).sample(5, axis=1).sample(10, axis=0)
```

	social_science	science	health_fitness	family_relationships	juvenile_nonfiction
title					
American Indian Myths and Legends (Pantheon Fairy Tale and Folklore Library)	1.0	0.0	0.0	0.0	0.0
Prayers for a Planetary Pilgrim a Personal Manual for Prayer and Ritual	0.0	0.0	0.0	0.0	0.0
Mr. Right, Right Now! : How a Smart Woman Can Land Her Dream Man in 6 Weeks	0.0	0.0	0.0	1.0	0.0
Flower Essences and Vibrational Healing	0.0	0.0	1.0	0.0	0.0
1001 Secrets for Windows Nt Registry	0.0	0.0	0.0	0.0	0.0
Predictions Library: Numerology	0.0	0.0	0.0	0.0	0.0

▼ Cosine Similarity

```
from sklearn.metrics.pairwise import cosine_similarity
```

```
# Menghitung cosine similarity pada matrix tf-idf
cosine_sim = cosine_similarity(tfidf_matrix)
cosine_sim
```

```
array([[1., 0., 0., ..., 0., 0., 0.],
        [0., 1., 0., ..., 0., 0., 0.],
        [0., 0., 1., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 1., 0., 0.],
        [0., 0., 0., ..., 0., 1., 0.],
        [0., 0., 0., ..., 0., 0., 1.]])
```

```
# Membuat dataframe dari variabel cosine_sim dengan baris dan kolom berupa nama resto
cosine_sim_df = pd.DataFrame(cosine_sim, index=book_new['title'], columns=book_new['title'])
print('Shape:', cosine_sim_df.shape)
```

```
# Melihat similarity matrix pada setiap resto
cosine_sim_df.sample(5, axis=1).sample(3, axis=0)
```

Shape: (13048, 13048)

	Cultural Diversity in Organizations: Theory, Research & Practice	Affluence: The A-To-Z Steps to a Richer Life	Mind and Morals: Essays on Ethics and Cognitive Science	Watercolor Color (Dorling Kindersley Art School)	From Survival to Recovery: Growing Up in an Alcoholic Home	
title						
title						
Everyday Object Lessons for Youth						
When Sparks						

▼ Mendapatkan rekomendasi

```
def book_recommendation(nama_buku, similarity_data=cosine_sim_df, items=book_new[['title', 'category']], k=5):
    # Mengambil data dengan menggunakan argpartition untuk melakukan partisi secara tidak langsung sepanjang sumbu yang dit
    # Dataframe diubah menjadi numpy
    # Range(start, stop, step)
    index = similarity_data.loc[:,nama_buku].to_numpy().argpartition(
        range(-1, -k, -1))

    # Mengambil data dengan similarity terbesar dari index yang ada
    closest = similarity_data.columns[index[-1:-(k+2):-1]]

    # Drop nama_resto agar nama resto yang dicari tidak muncul dalam daftar rekomendasi
    closest = closest.drop(nama_buku, errors='ignore')
    df = pd.DataFrame(closest).merge(items)
    df.drop_duplicates(keep='first', subset="title", inplace=True)
    return df.head(k)
```

book_new.head()

	title	author	category	publisher	
0	New Vegetarian: Bold and Beautiful Recipes for...	Celia Brooks Brown	Cooking	Ryland Peters & Small Ltd	
1	The Therapeutic Touch: How to Use Your Hands t...	Dolores Krieger	Health_Fitness	Fireside	
2	The Dragons of Eden: Speculations on the Evolu...	Carl Sagan	Science	Ballantine Books	
3	McDonald's: Behind the Arches	John F. Love	Business_Economics	Bantam	
4	Creating Wealth : Retire in Ten Years Using Al...	Robert G. Allen	Business_Economics	Fireside	

Next steps: [Generate code with book_new](#) [New interactive sheet](#)

book_new[book_new['title'].eq("McDonald's: Behind the Arches")]

	title	author	category	publisher	
3	McDonald's: Behind the Arches	John F. Love	Business_Economics	Bantam	

book_recommendation("McDonald's: Behind the Arches", k=10)

	title	category	
0	FUZZY LOGIC: THE REVOLUTIONARY COMPUTER TECHNO...	Business_Economics	
1	The Motley Fool Investment Guide: How the Fool...	Business_Economics	
2	Chicago's Museums: A Complete Guide to the Cit...	Business_Economics	
3	PASSION PROFIT POWER	Business_Economics	
4	If You Want to Be Rich & Happy: Don't Go to Sc...	Business_Economics	
5	Startup: A Silicon Valley Adventure	Business_Economics	
6	Staying Alive: Women, Ecology and Development	Business_Economics	
7	Leadership and the One Minute Manager : Increa...	Business_Economics	
8	The Job Hunter's Catalog	Business_Economics	
9	The Financially Confident Woman	Business_Economics	