```
import numpy as np
import pandas as pd
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
import matplotlib.pyplot as plt


# get data files
!wget https://cdn.freecodecamp.org/project-data/books/book-crossings.zip

!unzip book-crossings.zip

books_filename = 'BX-Books.csv'
ratings_filename = 'BX-Book-Ratings.csv'
```

```
--2024-10-30 17:55:42--  https://cdn.freecodecamp.org/project-data/books/book-crossings.
Resolving cdn.freecodecamp.org (cdn.freecodecamp.org)... 104.26.2.33, 172.67.70.149, 104
Connecting to cdn.freecodecamp.org (cdn.freecodecamp.org)|104.26.2.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26085508 (25M) [application/zip]
Saving to: ‘book-crossings.zip.2’

book-crossings.zip. 100%[===================>]  24.88M   126MB/s    in 0.2s

2024-10-30 17:55:42 (126 MB/s) - ‘book-crossings.zip.2’ saved [26085508/26085508]

Archive:  book-crossings.zip
replace BX-Book-Ratings.csv? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

```
# import csv data into dataframes
df_books = pd.read_csv(
    books_filename,
    encoding = "ISO-8859-1",
    sep=";",
    header=0,
    names=['isbn', 'title', 'author'],
    usecols=['isbn', 'title', 'author'],
    dtype={'isbn': 'str', 'title': 'str', 'author': 'str'})

df_ratings = pd.read_csv(
    ratings_filename,
    encoding = "ISO-8859-1",
    sep=";",
    header=0,
    names=['user', 'isbn', 'rating'],
    usecols=['user', 'isbn', 'rating'],
    dtype={'user': 'int32', 'isbn': 'str', 'rating': 'float32'})
```

Double-click (or enter) to edit

```
# add your code here - consider creating a new cell for each section of code
df_books.head()
df_ratings.head()
df_books.info()
df_ratings.user.unique()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271379 entries, 0 to 271378
Data columns (total 3 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   isbn    271379 non-null  object
 1   title   271379 non-null  object
 2   author  271377 non-null  object
dtypes: object(3)
memory usage: 6.2+ MB
array([276725, 276726, 276727, ..., 276709, 276721, 276723], dtype=int32)
```

```
df = df_ratings

counts1 = df['user'].value_counts()
counts2 = df['isbn'].value_counts()

df = df[~df['user'].isin(counts1[counts1 < 200].index)]
df = df[~df['isbn'].isin(counts2[counts2 < 100].index)]

df = pd.merge(right=df, left = df_books, on="isbn")

df = df.drop_duplicates(["title", "user"])

piv = df.pivot(index='title', columns='user', values='rating').fillna(0)


matrix = piv.values



matrix.shape
```

```
(673, 888)
```

```
piv
```

user	254	2276	2766	2977	3363	4017	4385	6242	6251	6323	...	274004	2740
title													
1st to Die: A Novel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
A Is for Alibi (Kinsey Millhone Mysteries (Paperback))	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
A Map of the World	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
A Painted House	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
A Prayer for Owen Meany	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	
Where the Heart Is (Oprah's Book Club (Paperback))	0.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
While I Was Gone	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	...	0.0	0
White Oleander : A Novel (Oprah's Book Club)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
Wicked: The Life and Times of the Wicked Witch of the West	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	0.0	0.0	...	0.0	0
Wild Animus	0.0	0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0

99 rows × 857 columns

```
from sklearn.neighbors import NearestNeighbors
model_knn=NearestNeighbors(metric='cosine',algorithm='brute')
model_knn.fit(matrix)
```

```
NearestNeighbors
NearestNeighbors(algorithm='brute', metric='cosine')
```

```
get_recommends('The Queen of the Damned (Vampire Chronicles (Paperback))')
```

```
['The Queen of the Damned (Vampire Chronicles (Paperback))',
 [['Catch 22', 0.7939835419270879],
  ['The Witching Hour (Lives of the Mayfair Witches)', 0.7448657003312193],
  ['Interview with the Vampire', 0.7345068863988313],
  ['The Tale of the Body Thief (Vampire Chronicles (Paperback))',
   0.5376338446489461],
  ['The Vampire Lestat (Vampire Chronicles, Book II)', 0.5178411864186412],
  ['The Queen of the Damned (Vampire Chronicles (Paperback))',
   1.1102230246251565e-16]]]
```

```
books = get_recommends("Where the Heart Is (Oprah's Book Club (Paperback))")
print(books)

def test_book_recommendation():
  test_pass = True
  recommends = get_recommends("Where the Heart Is (Oprah's Book Club (Paperback))")
  if recommends[0] != "Where the Heart Is (Oprah's Book Club (Paperback))":
    test_pass = False
  recommended_books = ["I'll Be Seeing You", 'The Weight of Water', 'The Surgeon', 'I Know T
  recommended_books_dist = [0.8, 0.77, 0.77, 0.77]
  for i in range(2):
    if recommends[1][i][0] not in recommended_books:
      test_pass = False
    if abs(recommends[1][i][1] - recommended_books_dist[i]) >= 0.05:
      test_pass = False
  if test_pass:
    print("You passed the challenge! 🎉🎉🎉🎉🎉")
  else:
    print("You haven't passed yet. Keep trying!")

test_book_recommendation()
```

```
["Where the Heart Is (Oprah's Book Club (Paperback))", [["I'll Be Seeing You", 0.801621
You passed the challenge! 🎉🎉🎉🎉🎉
```