Product Recommendation System based on Amazon Review

Mahisha Ramesh - MT23121

Questions

2) Choose a product of your choice. Let's say ‘Headphones’.

```
# Function to process and write a chunk of data to CSV
def process_chunk(chunk, csv_writer):
    for data in chunk:
        if 'Electronics' in data.get('category', []) and 'Headphones' in data.get('category'
            csv_writer.writerow(data)

# Open the input JSON file
with open('/kaggle/input/meta-electronics-dataset/meta_Electronics.json', 'r', encoding='utf
    # Initialize CSV writer and open output CSV file
```

Output:-

["['Electronics', 'Headphones']", '', "['Use these high quality headphones for internet chatting and enjoy the comfort and ease of the headphones with the microphone and in-line volume control.Works with: Skype msn AIM YAHOO! Windows Live']", '', 'Polaroid Pbm2200 PC / Gaming Stereo Headphones With Microphone & In-line Volume', '[]', '', 'Polaroid', "['Ideal for PC Internet chatting, PC / Console gaming and music', 'In-line volume control', 'Optimal performance for VoIP usage', 'Enhanced soft-cushioned ear pads']", "['>#3,548,269 in Cell Phones & Accessories (See Top 100 in Cell Phones & Accessories)', '>#122,201 in Cell Phones & Accessories > Cell Phone Accessories > Headphones', '>#366,901 in Electronics > Home Audio & Theater', '>#387,499 in Electronics > Portable Audio & Video > MP3 & MP4 Player Accessories']", '[]', 'All Electronics', '', 'December 13, 2012', '', '0558835155', "['https://images-na.ssl-images-amazon.com/images/I/21rEirndRLL._SS40_.jpg']", "['https://images-na.ssl-images-amazon.com/images/I/21rEirndRLL.jpg']"]
["['Electronics', 'Headphones', 'Earbud Headphones']", '', "['Barnes and noble official nook earphones.']", '', 'Official Nook Audio Ie250 Earphones', '[]', '', 'Nook', '[]', "['>#4,167,961 in Cell Phones & Accessories (See Top 100 in Cell Phones & Accessories)', '>#50,473 in Cell Phones & Accessories > Cell Phone Accessories > Headphones > Earbud Headphones', '>#403,963 in Electronics > Home Audio & Theater', '>#427,806 in Electronics > Portable Audio & Video > MP3 & MP4 Player Accessories']", '[]', 'Home Audio & Theater', '', 'September 18, 2013', '', '0594478162', "['https://images-na.ssl-images-amazon.com/images/I/41-ZZ1e7OEL._SS40_.jpg']", "['https://images-na.ssl-images-amazon.com/images/I/41-ZZ1e7OEL.jpg']"]

3)Report the total number of rows for the product. Perform appropriate pre-processing as handling missing values, duplicates and other

```
Number of rows after processing: 31115
Processed data saved to processed_headphone.csv
```

4. Obtain the Descriptive Statistics of the product as : -

```
a. Number of Reviews: 32908
b. Average Rating Score: 3.814239698553543
c. Number of Unique Products: 3413
d. Number of Good Ratings: 26018
e. Number of Bad Ratings: 6890
f. Number of Reviews corresponding to each Rating:
overall
1.0     4069
2.0     2821
3.0     3839
4.0     6604
5.0    15575
Name: count, dtype: int64
```

5) Preprocess the Text

```python
# Define a function to preprocess text combining all the steps
def preprocess_text(text):
    text = remove_html_tags(text)
    text = remove_accented_chars(text)
    text = expand_headphones_acronyms(text)
    text = remove_special_characters(text)
    text = remove_punctuation_and_stopwords(text)
    text = lemmatize_text_with_spacy(text)  # Use spaCy for lemmatization
    return text
```

6)To extract relevant statistics, perform the following EDA -
a. Top 20 most reviewed brands in the category that you have chosen.

```
brand
Sony                  4864
Sennheiser            4698
Koss                  1759
Audio-Technica        1169
Bose                  1140
JVC                   1044
Philips                906
Panasonic              800
beyerdynamic           716
Shure                  675
Klipsch                673
V-MODA                 668
Plantronics            659
Skullcandy             563
JLAB                   523
Beats                  508
Etymotic Research      441
AKG                    424
Ultimate Ears          413
Monster                355
Name: count, dtype: int64
```

b. Top 20 least reviewed brands in the category you have chosen.

```
brand
TYLT           1
BLUETTEK       1
Spark          1
Pengaz         1
Doosl          1
netjnp         1
ECCPE          1
SeattleTech    1
Haldirect      1
jarv           1
Fonus          1
Gearonic TM    1
Blackcell      1
Sades          1
fitTek         1
KickBot        1
Sunweb         1
HeadGear       1
Extreme 80s    1
Sunnice        1
Name: count, dtype: int64
```

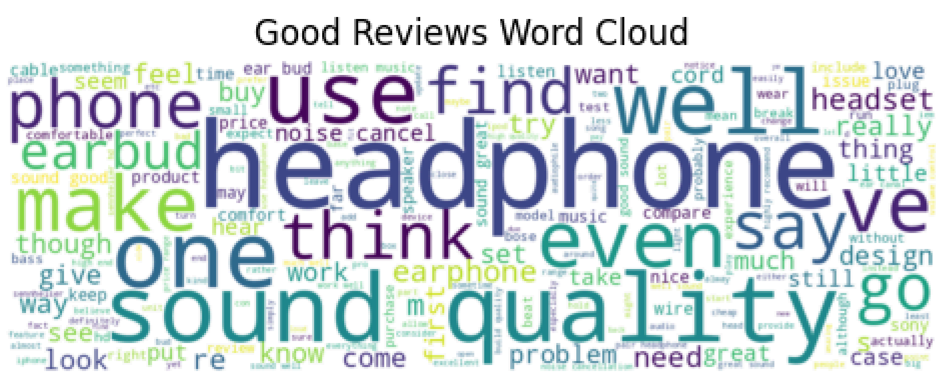
c. Which is the most positively reviewed ‘Headphone’ (Or for any other electronic product you have selected)

Most positively reviewed ASIN ID: B00004TZJI

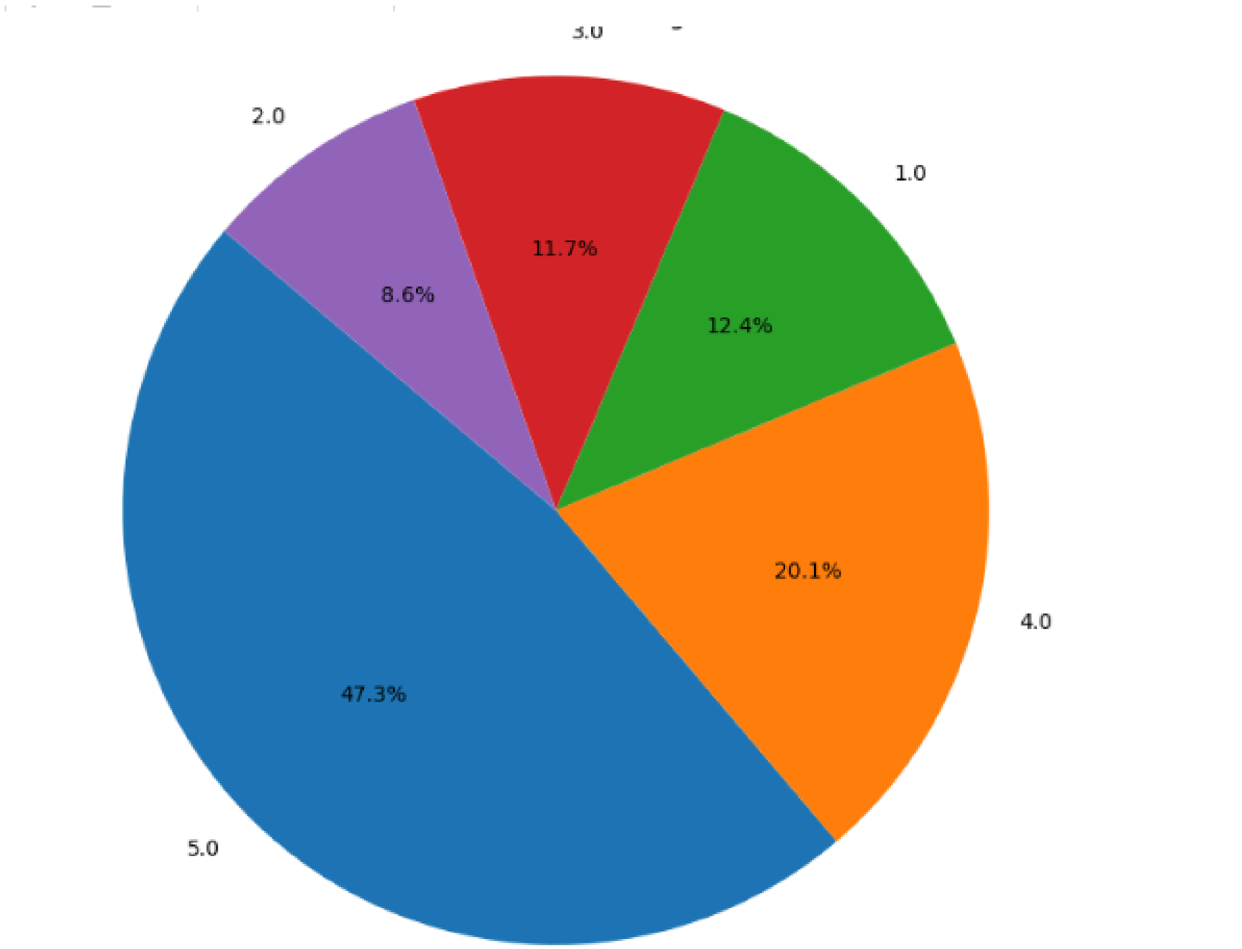
d.Show the count of ratings for the product over 5 consecutive years.


```
[{'asin': 'B000001OMI', 'year_range': '2007-2011', 'count': 2}, {'asin': 'B000001OMI', 'year_range': '2008-20
12', 'count': 2}, {'asin': 'B000001OMR', 'year_range': '2005-2009', 'count': 1}, {'asin': 'B000001OMR', 'year
_range': '2006-2010', 'count': 0}, {'asin': 'B000001OMR', 'year_range': '2007-2011', 'count': 0}, {'asin': 'B
000001OMR', 'year_range': '2008-2012', 'count': 0}, {'asin': 'B000001OMR', 'year_range': '2009-2013', 'coun
t': 0}, {'asin': 'B000001OMR', 'year_range': '2010-2014', 'count': 0}, {'asin': 'B000001OMR', 'year_range':
'2011-2015', 'count': 0}, {'asin': 'B00000J1EJ', 'year_range': '2001-2005', 'count': 4}, {'asin': 'B00000J1E
J', 'year_range': '2002-2006', 'count': 5}, {'asin': 'B00000J1EJ', 'year_range': '2003-2007', 'count': 6},
{'asin': 'B00000J1EJ', 'year_range': '2004-2008', 'count': 13}, {'asin': 'B00000J1EJ', 'year_range': '2005-20
09', 'count': 14}, {'asin': 'B00000J1EJ', 'year_range': '2006-2010', 'count': 14}, {'asin': 'B00000J1EJ', 'ye
ar_range': '2007-2011', 'count': 12}, {'asin': 'B00000J1EJ', 'year_range': '2008-2012', 'count': 10}, {'asi
n': 'B00000J1EJ', 'year_range': '2009-2013', 'count': 3}, {'asin': 'B00000J1EJ', 'year_range': '2010-2014',
'count': 1}, {'asin': 'B00000JBHP', 'year_range': '2010-2014', 'count': 3}, {'asin': 'B00000JBHP', 'year_rang
e': '2011-2015', 'count': 2}, {'asin': 'B00000JBHP', 'year_range': '2012-2016', 'count': 4}, {'asin': 'B00001
P4XA', 'year_range': '2000-2004', 'count': 18}, {'asin': 'B00001P4XA', 'year_range': '2001-2005', 'count': 2
7}, {'asin': 'B00001P4XA', 'year_range': '2002-2006', 'count': 46}, {'asin': 'B00001P4XA', 'year_range': '200
3-2007', 'count': 65}, {'asin': 'B00001P4XA', 'year_range': '2004-2008', 'count': 66}, {'asin': 'B00001P4XA',
'year_range': '2005-2009', 'count': 63}, {'asin': 'B00001P4XA', 'year_range': '2006-2010', 'count': 52}, {'as
in': 'B00001P4XA', 'year_range': '2007-2011', 'count': 33}, {'asin': 'B00001P4XA', 'year_range': '2008-2012',
'count': 13}, {'asin': 'B00001P4XA', 'year_range': '2009-2013', 'count': 14}, {'asin': 'B00001P4XA', 'year_ra
nge': '2010-2014', 'count': 22}, {'asin': 'B00001P4XA', 'year_range': '2011-2015', 'count': 21}, {'asin': 'B0
0001P4XA', 'year_range': '2012-2016', 'count': 20}, {'asin': 'B00001P4XH', 'year_range': '2000-2004', 'coun
t': 1}, {'asin': 'B00001P4XH', 'year_range': '2001-2005', 'count': 1}, {'asin': 'B00001P4XH', 'year_range':
'2002-2006', 'count': 1}, {'asin': 'B00001P4XH', 'year_range': '2003-2007', 'count': 6}, {'asin': 'B00001P4X
H', 'year_range': '2004-2008', 'count': 12}, {'asin': 'B00001P4XH', 'year_range': '2005-2009', 'count': 24},
```


e. Form a Word Cloud for 'Good' and 'Bad' ratings. Report the most commonly used words for positive and negative reviews by observing the good and bad word clouds.



F. Plot a pie chart for Distribution of Ratings vs. the No. of Reviews.



G. Report in which year the product got maximum reviews.

In the year 2014, the product received the maximum reviews: 6217 reviews.

h. Which year has the highest number of Customers?

In the year 2014, there were the highest number of customers: 5434 customers.

year

2000 15

2001 29

2002 52

2003	84
2004	203
2005	564
2006	896
2007	1323
2008	1346
2009	1466
2010	1550
2011	2141
2012	3017
2013	4544
2014	5434
2015	2921
2016	1847
2017	906
2018	99

Name: reviewerID, dtype: int64

7. Use a relevant feature engineering technique to model review text as Bag of Words model, TF-IDF, Hashing Vectorizer or Word2Vec.

[{'buy': 0.04828998481044271, 'son': 0.17756230619433183, 'want': 0.06273959473973903, 'headphone': 0.021935256490506358, 'go': 0.05092156185353742, 'ear': 0.030964737705080004, 'like': 0.03552669075710617, 'school': 0.20153581223198025, 'good': 0.05318245282440118, 'price': 0.09730947530087833, 'really': 0.05424879540088827, 'sound': 0.013413511063846336, 'quality': 0.036306258242803174, 'horrible': 0.17116502591848906, 'wiggle': 0.23356622313722825, 'cord': 0.07448352548448908, 'often': 0.135838949105822, 'package': 0.13630339128186006, 'say': 0.060080823219720386, 'wow': 0.16699121445581613, 'glad': 0.16326135622637586, 'break': 0.0889994570849165}

8.Compare the performance of 5 Machine Learning based models on the basis of

Precision, Recall, F-1 Score and Support for each of the 3 target classes distinctly.

1) Random Forest

Classification Report:

	precision	recall	f1-score	support
0	0.93	0.91	0.92	5491
1	0.81	0.85	0.83	5570
2	0.82	0.79	0.81	5572
accuracy			0.85	16633
macro avg	0.85	0.85	0.85	16633
weighted avg	0.85	0.85	0.85	16633

2) SUPPORT VECTOR CLASSIFIER¶

Classification Report:

	precision	recall	f1-score	support
0	0.44	0.47	0.45	869
1	0.39	0.57	0.46	827
2	0.43	0.20	0.27	804
accuracy			0.41	2500
macro avg	0.42	0.41	0.40	2500
weighted avg	0.42	0.41	0.40	2500

3) GRADIENT BOOSTING

Classification Report:

	precision	recall	f1-score	support
0	0.61	0.65	0.63	1244
1	0.57	0.59	0.58	1257
2	0.61	0.54	0.57	1249
accuracy			0.60	3750
macro avg	0.60	0.60	0.59	3750
weighted avg	0.60	0.60	0.59	3750

4) DECISION TREE

```
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.54      0.54      1303
           1       0.46      0.50      0.48      1205
           2       0.46      0.40      0.43      1242

    accuracy                           0.48      3750
   macro avg       0.48      0.48      0.48      3750
weighted avg       0.48      0.48      0.48      3750
```

5) KNN

```
Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.83      0.60      1229
           1       0.55      0.63      0.58      1250
           2       0.57      0.07      0.13      1271

    accuracy                           0.51      3750
   macro avg       0.53      0.51      0.44      3750
weighted avg       0.53      0.51      0.44      3750
```

11) Collaborative Filtering :

a) Create a user-item rating matrix

```
asin           B000001OMI  B00000JCTO  B00001P4XA  B00001P4XH  B00001P4YG  \
reviewerID
A118GK08650JY7        0.0         0.0         0.0         0.0         0.0
A12DQZKRKTNF5E        0.0         0.0         0.0         0.0         0.0
A13WL1MBY347F7        0.0         0.0         0.0         0.0         0.0
A149RNR5RH19YY        0.0         0.0         0.0         0.0         0.0
A166PLPFD2A42H        0.0         0.0         0.0         0.0         0.0
...                   ...         ...         ...         ...         ...
AWPODHOB4GFWL         0.0         0.0         0.0         0.0         0.0
AXU3VKZE848IY         0.0         0.0         0.0         0.0         0.0
AYTGG6XTVUG7G         0.0         0.0         0.0         0.0         0.0
AZ0SIZRQWN7RC         0.0         0.0         0.0         0.0         0.0
AZXFS8GCTSQ5R         0.0         0.0         0.0         0.0         0.0

asin           B00001P4ZH  B00001P505  B00001W0ET  B00001WRSJ  B00004SD88  \
reviewerID
A118GK08650JY7        0.0         0.0         0.0         0.0         0.0
A12DQZKRKTNF5E        0.0         0.0         0.0         0.0         0.0
A13WL1MBY347F7        0.0         0.0         0.0         0.0         0.0
A149RNR5RH19YY        5.0         0.0         0.0         0.0         0.0
A166PLPFD2A42H        0.0         0.0         0.0         0.0         0.0
```

c) Create a user-user recommender system - i.e,
i) Find the top N similar users, by using cosine similarity. N = 10, 20, 30, 40, 50

```
       ReviewerID       Nearest_1       Nearest_2       Nearest_3  \
0   A103OG69UZTK9   AABY9VMRDRFFE  A25KAWJE80DHJZ  A1E1BGAH9X4MS3
1  A118GK08650JY7  A2X9SQSKV4WKT0  A37PZJH2F13IOR  A2SHUDXDYMS9MA
2  A11MTYZ120N08D  A281T8MXBOSWY5  A3MTKYOAMWJE9B  A3N4VTNFPMTHEF
3  A11X9HWN09P7MC  A3MKAP4NUTYKMN  A3MUO47CT6EQF8  A3KKM0T1KY42HA
4   A11ZYI5IG7V0O   ATS2855497V0I  A3LKN9GND01EWJ  A3MKAP4NUTYKMN

        Nearest_4       Nearest_5       Nearest_6       Nearest_7  \
0  A1PM7HH4F77NEH  A3AH7GTE88QNPL   ALZJMBRRKUEON   A2KN1ILG8TABA
1  A1HM9SAU8TV134  A21KNRUAA5RK5E  A3EJDV2RTFROU4  A2CW9GKMNFAU6R
2   A3L1EH8KUWVZB  A3LGT6UZL99IW1  A3LKN9GND01EWJ  A3LWC833HQIG7J
3   A3L1EH8KUWVZB  A3LGT6UZL99IW1  A3LKN9GND01EWJ  A3LWC833HQIG7J
4  A3MUO47CT6EQF8  A3KKM0T1KY42HA   A3L1EH8KUWVZB  A3LGT6UZL99IW1

        Nearest_8       Nearest_9      Nearest_10
0  A28XMCDOV3QUTJ   A4M61F235UDNL   AMS7E10YOOUY0
1  A1YAGM2QOSAAOT   AYUF7YETYOLNX  A17HMM1M7T9PJ1
2  A3MKAP4NUTYKMN  A3K91X9X2ARDOK  A3NCIN6TNL0MGA
3  A3K6J60D3LX28Q  A3MTKYOAMWJE9B  A3NCIN6TNL0MGA
4  A3LWC833HQIG7J  A3K6J60D3LX28Q  A3NCIN6TNL0MGA
```

b) Normalize the ratings, by using min-max scaling on user's reviews

```
asin             B000001OMI  B00000JCTO  B00001P4XA  B00001P4XH  B00001P4YG  \
reviewerID
A118GK08650JY7     3.996521    3.996521    3.996521    3.996521    3.996521
A12DQZKRKTNF5E     3.996521    3.996521    3.996521    3.996521    3.996521
A13WL1MBY347F7     3.996521    3.996521    3.996521    3.996521    3.996521
A149RNR5RH19YY     3.900000    3.900000    3.900000    3.900000    3.900000
A166PLPFD2A42H     4.300000    4.300000    4.300000    4.300000    4.300000
...                     ...         ...         ...         ...         ...
AWPODHOB4GFWL      4.100000    4.100000    4.100000    4.100000    4.100000
AXU3VKZE848IY      4.100000    4.100000    4.100000    4.100000    4.100000
AYTGG6XTVUG7G      4.500000    4.500000    4.500000    4.500000    4.500000
AZ0SIZRQWN7RC      3.900000    3.900000    3.900000    3.900000    3.900000
AZXFS8GCTSQ5R      3.996521    3.996521    3.996521    3.996521    3.996521
```

```
                B000001OMI  B00000JCTO  B00001P4XA  B00001P4XH  B00001P4YG  \
reviewerID
A118GK08650JY7    0.522153    0.609179     0.74913    0.739452    0.609179
A12DQZKRKTNF5E    0.522153    0.609179     0.74913    0.739452    0.609179
A13WL1MBY347F7    0.522153    0.609179     0.74913    0.739452    0.609179
A149RNR5RH19YY    0.476190    0.555556     0.72500    0.703704    0.555556
A166PLPFD2A42H    0.666667    0.777778     0.82500    0.851852    0.777778
...                    ...         ...         ...         ...         ...
AWPODHOB4GFWL     0.571429    0.666667     0.77500    0.777778    0.666667
AXU3VKZE848IY     0.571429    0.666667     0.77500    0.777778    0.666667
AYTGG6XTVUG7G     0.761905    0.888889     0.87500    0.925926    0.888889
AZ0SIZRQWN7RC     0.476190    0.555556     0.72500    0.703704    0.555556
AZXFS8GCTSQ5R     0.522153    0.609179     0.74913    0.739452    0.609179

                B00001P4ZH  B00001P505  B00001WRSJ  B00004SD88  B00004SY4H  \
reviewerID
A118GK08650JY7    0.522153    0.522153    0.665507    0.609179    0.522153
A12DQZKRKTNF5E    0.522153    0.522153    0.665507    0.609179    0.522153
A13WL1MBY347F7    0.522153    0.522153    0.665507    0.609179    1.000000
A149RNR5RH19YY    1.000000    0.476190    0.633333    0.555556    0.476190
```

Normalized ratings between 0 and 1

c) ii) Use K-folds validation. K = 5. Explanation: Create 5 subsets, and take 1 of them as the validation set. Take the rest 4 to be the training set.
iii) Use the training set to predict the missing values, and use the validation set to calculate the error. (Error = |actual_rating - predicted_rating|)

Code:-

```python
# Define the number of folds
k_folds = 5

# Initialize KFold cross-validator
kf = KFold(n_splits=k_folds)

# Initialize list to store errors for each fold
errors = []

# Iterate over folds
for train_index, val_index in kf.split(user_item_matrix):
    # Split data into training and validation sets
    train_set = user_item_matrix.iloc[train_index]
    val_set = user_item_matrix.iloc[val_index]

    # Predict missing values using training set
    predicted_ratings = train_set.mean(axis=0)  # Use mean ratings as predictions

    # Calculate error for the validation set
    error = np.abs(val_set - predicted_ratings)
```

Average error across 5 folds: 0.1043198605448111

d) Create an item-item recommender system. Use the same steps as above.

Item-item Similarity

```
            B000001OMI  B00000JCTO  B00001P4XA  B00001P4XH  B00001P4YG  \
B000001OMI    1.000000   -0.005682   -0.011913   -0.005682   -0.005682
B00000JCTO   -0.005682    1.000000    0.484234   -0.005682   -0.005682
B00001P4XA   -0.011913    0.484234    1.000000   -0.011913   -0.011913
B00001P4XH   -0.005682   -0.005682   -0.011913    1.000000   -0.005682
B00001P4YG   -0.005682   -0.005682   -0.011913   -0.005682    1.000000
...                ...         ...         ...         ...         ...
B00KRPKIYQ   -0.005682   -0.005682   -0.011913   -0.005682   -0.005682
B00KTCMJKI   -0.005682   -0.005682   -0.011913   -0.005682   -0.005682
B00KVTOOUW   -0.005682   -0.005682   -0.011913   -0.005682   -0.005682
B00KWKKG6G   -0.005682   -0.005682   -0.011913   -0.005682   -0.005682
B00KWMNDDM   -0.005682   -0.005682   -0.011913   -0.005682   -0.005682

            B00001P4ZH  B00001P505  B00001W0ET  B00001WRSJ  B00004SD88  ...  \
B000001OMI    -0.02092   -0.011413    1.000000   -0.015768   -0.005682  ...
B00000JCTO    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
B00001P4XA     0.34316   -0.023930   -0.011913   -0.033060   -0.011913  ...
B00001P4XH    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
B00001P4YG    -0.02092    0.413883   -0.005682    0.333092   -0.005682  ...
...                ...         ...         ...         ...         ...  ...
B00KRPKIYQ    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
B00KTCMJKI    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
B00KVTOOUW    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
B00KWKKG6G    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
B00KWMNDDM    -0.02092   -0.011413   -0.005682   -0.015768   -0.005682  ...
```