SNAP

April 22, 2024

[22]:
```
import os
import tempfile
import shutil
import urllib
import zipfile
import pandas as pd
```

1.

[23]:
```
import pandas as pd

#  CSV
def read_snap_dataset(file_path):
    #  CSV       SOURCE, TARGET, RATING, TIME
    try:
        df = pd.read_csv(file_path, header=None, names=["SOURCE", "TARGET",␣
 ↪"RATING", "TIME"])
        return df
    except FileNotFoundError:
        print("          ")

#  CSV
file_path = "./input/soc-sign-bitcoinalpha.csv"

#
snap_dataset = read_snap_dataset(file_path)

snap_dataset
```

[23]:

	SOURCE	TARGET	RATING	TIME
0	7188	1	10	1407470400
1	430	1	10	1376539200
2	3134	1	10	1369713600
3	3026	1	10	1350014400
4	3010	1	10	1347854400
…	…	…	…	…
24181	7604	7601	10	1364270400
24182	7601	7604	10	1364270400

```
24183    7604    7602      10  1364270400
24184    7602    7604      10  1364270400
24185    7604    7603     -10  1364270400

[24186 rows x 4 columns]
```

[24]:
```
snap_dataset = snap_dataset.dropna()  #
snap_dataset
```

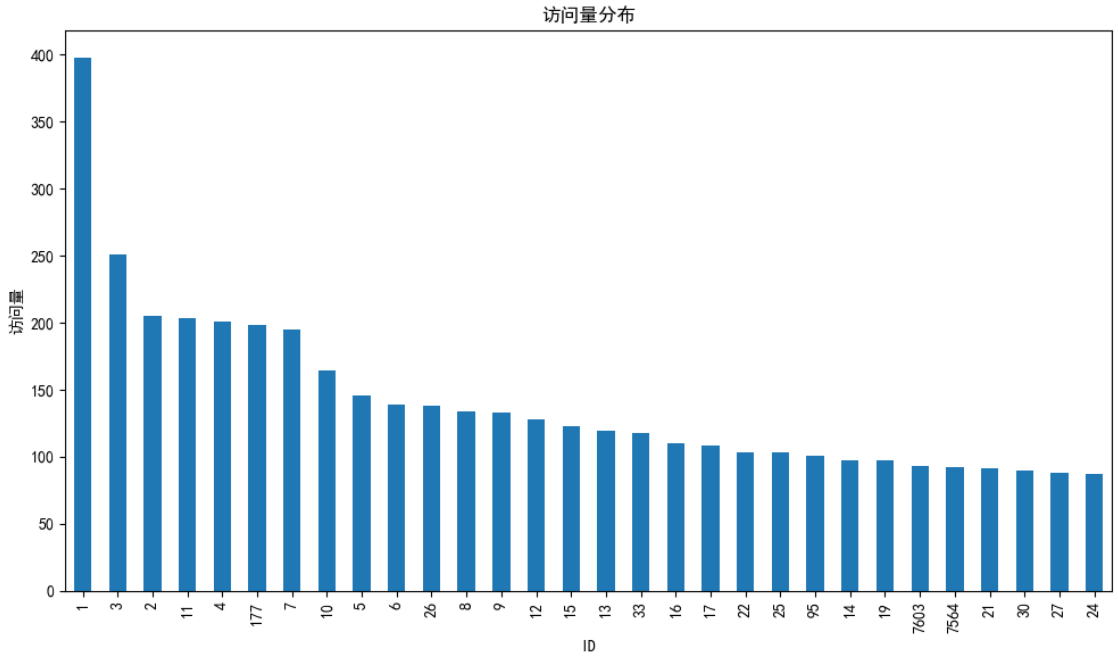
[24]:
```
        SOURCE  TARGET  RATING        TIME
0         7188       1      10  1407470400
1          430       1      10  1376539200
2         3134       1      10  1369713600
3         3026       1      10  1350014400
4         3010       1      10  1347854400
…          …     …       …          …
24181    7604    7601      10  1364270400
24182    7601    7604      10  1364270400
24183    7604    7602      10  1364270400
24184    7602    7604      10  1364270400
24185    7604    7603     -10  1364270400

[24186 rows x 4 columns]
```

[25]:
```
#
page_visits = snap_dataset['TARGET'].value_counts()

#
most_visited_pages = page_visits.head(30)

#
print("     ")
print(most_visited_pages)
```

```
TARGET
1       398
3       251
2       205
11      203
4       201
177     198
7       195
10      164
5       146
6       139
26      138
8       134
```

```
9       133
12      128
15      123
13      119
33      118
16      110
17      108
22      103
25      103
95      101
14       97
19       97
7603     93
7564     92
21       91
30       90
27       88
24       87
Name: count, dtype: int64
```

[26]:
```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']  #       SimHei

#
plt.figure(figsize=(10, 6))
most_visited_pages.plot(kind='bar')
plt.xlabel("ID")
plt.ylabel("  ")
plt.title("   ")
plt.tight_layout()  #
plt.show()
```



30 “ ” Long Tail Distribution

3. : Apriori FP-growth Apriori

[30]:
```
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
#      Apriori
user_data = []
last_user = '24186'
tmp = []

for index, row in snap_dataset.iterrows():
    user_id = row['SOURCE']
    vroot_id = row['TARGET']

    if user_id == last_user:
        tmp.append(vroot_id)
    else:
        user_data.append(tmp)
        tmp = []
        tmp.append(vroot_id)
    last_user = user_id

user_data.append(tmp)
# user_data
```

```
te = TransactionEncoder()
data_encoded = te.fit_transform(user_data)
df = pd.DataFrame(data_encoded, columns=te.columns_)

df
```

[30]:

```
         1      2      3      4      5      6      7      8      9      10    \
0      False  False  False  False  False  False  False  False  False  False
1       True  False  False  False  False  False  False  False  False  False
2       True  False  False  False  False  False  False  False  False  False
3       True  False  False  False  False  False  False  False  False  False
4       True  False  False  False  False  False  False  False  False  False
…       …     …      …      …      …      …      …      …      …      …
12869  False  False  False  False  False  False  False  False  False  False
12870  False  False  False  False  False  False  False  False  False  False
12871  False  False  False  False  False  False  False  False  False  False
12872  False  False  False  False  False  False  False  False  False  False
12873  False  False  False  False  False  False  False  False  False  False

       …   7595   7596   7597   7598   7599   7600   7601   7602   7603  \
0      …  False  False  False  False  False  False  False  False  False
1      …  False  False  False  False  False  False  False  False  False
2      …  False  False  False  False  False  False  False  False  False
3      …  False  False  False  False  False  False  False  False  False
4      …  False  False  False  False  False  False  False  False  False
…    …    …      …      …      …      …      …      …      …      …
12869  …  False  False  False  False  False  False   True  False  False
12870  …  False  False  False  False  False  False  False  False  False
12871  …  False  False  False  False  False  False  False   True  False
12872  …  False  False  False  False  False  False  False  False  False
12873  …  False  False  False  False  False  False  False  False   True

        7604
0      False
1      False
2      False
3      False
4      False
…      …
12869  False
12870   True
12871  False
12872   True
12873  False

[12874 rows x 3754 columns]
```

[47]:
```
#  Apriori
frequent_itemsets = apriori(df, min_support=0.01, use_colnames=True)

#
print("Frequent Itemsets:")
# print(frequent_itemsets)
frequent_itemsets
```

```
Frequent Itemsets:
```

[47]:

	support	itemsets
0	0.030915	(1)
1	0.015924	(2)
2	0.019497	(3)
3	0.015613	(4)
4	0.011341	(5)
..	…	…
437	0.001010	(7602, 7604, 7598)
438	0.001010	(7601, 7604, 7599)
439	0.001010	(7602, 7604, 7599)
440	0.001010	(7601, 7604, 7598, 7599)
441	0.001010	(7602, 7604, 7598, 7599)

[442 rows x 2 columns]

[49]:
```
rules = association_rules(frequent_itemsets, metric="confidence",␣
 ↪min_threshold=0.3)

#
print("\nAssociation Rules:")
# print(rules)
rules
```

```
Association Rules:
```

[49]:

	antecedents	consequents	antecedent support	consequent support	\
0	(47)	(31)	0.003418	0.005593	
1	(47)	(145)	0.003418	0.006525	
2	(136)	(177)	0.002563	0.015380	
3	(7595)	(177)	0.003651	0.015380	
4	(7598)	(7599)	0.001476	0.001476	
..	…	…	…	…	
79	(7604, 7599)	(7602, 7598)	0.001243	0.001010	
80	(7598, 7599)	(7602, 7604)	0.001165	0.001165	
81	(7602)	(7604, 7598, 7599)	0.001320	0.001165	
82	(7598)	(7602, 7604, 7599)	0.001476	0.001010	
83	(7599)	(7602, 7604, 7598)	0.001476	0.001010	

```
     support  confidence        lift  leverage  conviction  zhangs_metric
0   0.001243    0.363636   65.020202  0.001224    1.562640       0.987997
1   0.001087    0.318182   48.765152  0.001065    1.457097       0.982853
2   0.001010    0.393939   25.614019  0.000970    1.624623       0.963428
3   0.001243    0.340426   22.134537  0.001187    1.492811       0.958320
4   0.001165    0.789474  534.930748  0.001163    4.742990       0.999606
..       …           …          …         …           …              …
79  0.001010    0.812500  804.625000  0.001009    5.327948       1.000000
80  0.001010    0.866667  743.831111  0.001008    7.491261       0.999821
81  0.001010    0.764706  656.321569  0.001008    4.245048       0.999797
82  0.001010    0.684211  677.578947  0.001008    3.163469       1.000000
83  0.001010    0.684211  677.578947  0.001008    3.163469       1.000000

[84 rows x 10 columns]
```

[50]:
```
#
rules['lift'] = rules['lift'].apply(lambda x: round(x, 2))

#
observed = rules['support'] * len(df)  #
expected = rules['antecedent support'] * rules['consequent support'] * len(df)
 ↪#
chi_squared = ((observed - expected) ** 2 / expected).sum()  #

#
print("    :")
print(rules[['antecedents', 'consequents', 'lift']])
print("\n    :")
print("  :", chi_squared)
```

```
    :
     antecedents         consequents    lift
0           (47)                (31)   65.02
1           (47)               (145)   48.77
2          (136)               (177)   25.61
3         (7595)               (177)   22.13
4         (7598)              (7599)  534.93
..           …                   …     …
79  (7604, 7599)        (7602, 7598)  804.62
80  (7598, 7599)        (7602, 7604)  743.83
81        (7602)  (7604, 7598, 7599)  656.32
82        (7598)  (7602, 7604, 7599)  677.58
83        (7599)  (7602, 7604, 7598)  677.58

[84 rows x 3 columns]

   :
```

: 626814.4944881184

4. ”support” “confidence” ”lift”

0 (47) (31) 65.02 1 (47) (145) 48.77 2 (136) (177) 25.61 3 (7595) (177) 22.13

[ ]:

Microsoft

April 22, 2024

[1]:
```
import os
import tempfile
import shutil
import urllib
import zipfile
import pandas as pd
```

1.

[2]:
```
behaviors_path = os.path.join('./input', 'behaviors.tsv')
behaviors_df=pd.read_table(
    behaviors_path,
    header=None,
    names=['impression_id', 'user_id', 'time', 'history', 'impressions'])
behaviors_df
```

[2]:
```
        impression_id user_id                    time  \
0                   1  U13740   11/11/2019 9:05:58 AM
1                   2  U91836   11/12/2019 6:11:30 PM
2                   3  U73700   11/14/2019 7:01:48 AM
3                   4  U34670   11/11/2019 5:28:05 AM
4                   5   U8125   11/12/2019 4:11:21 PM
…                 …      …                       …
156960         156961  U21593  11/14/2019 10:24:05 PM
156961         156962  U10123   11/13/2019 6:57:04 AM
156962         156963  U75630  11/14/2019 10:58:13 AM
156963         156964  U44625   11/13/2019 2:57:02 PM
156964         156965  U64800   11/14/2019 3:25:49 PM

                                                 history  \
0       N55189 N42782 N34694 N45794 N18445 N63302 N104…
1       N31739 N6072 N63045 N23979 N35656 N43353 N8129…
2       N10732 N25792 N7563 N21087 N41087 N5445 N60384…
3       N45729 N2203 N871 N53880 N41375 N43142 N33013 …
4                           N10078 N56514 N14904 N33740
…                                                     …
156960  N7432 N58559 N1954 N43353 N14343 N13008 N28833…
```

```
156961  N9803 N104 N24462 N57318 N55743 N40526 N31726 …
156962  N29898 N59704 N4408 N9803 N53644 N26103 N812 N…
156963  N4118 N47297 N3164 N43295 N6056 N38747 N42973 …
156964                                        N22997 N48742

                                                impressions
0                                           N55689-1 N35729-0
1       N20678-0 N39317-0 N58114-0 N20495-0 N42977-0 N…
2       N50014-0 N23877-0 N35389-0 N49712-0 N16844-0 N…
3                     N35729-0 N33632-0 N49685-1 N27581-0
4       N39985-0 N36050-0 N16096-0 N8400-1 N22407-0 N6…
…                                                      …
156960  N2235-0 N22975-0 N64037-0 N47652-0 N11378-0 N4…
156961  N3841-0 N61571-0 N58813-0 N28213-0 N4428-0 N25…
156962  N55913-0 N62318-0 N53515-0 N10960-0 N9135-0 N5…
156963  N6219-0 N3663-0 N31147-0 N58363-0 N4107-0 N457…
156964                N61233-0 N33828-1 N19661-0 N41934-0

[156965 rows x 5 columns]
```

[3]:
```
news_path = os.path.join('./input', 'news.tsv')
news_df=pd.read_table(news_path,
              header=None,
              names=[
                  'id', 'category', 'subcategory', 'title', 'abstract', 'url',
                  'title_entities', 'abstract_entities'
              ])
news_df
```

[3]:
```
           id   category        subcategory  \
0      N55528  lifestyle    lifestyleroyals
1      N19639     health         weightloss
2      N61837       news          newsworld
3      N53526     health             voices
4      N38324     health            medical
…       …        …                 …
51277  N16909    weather  weathertopstories
51278  N47585  lifestyle    lifestylefamily
51279   N7482     sports        more_sports
51280  N34418     sports         soccer_epl
51281  N44276      autos        autossports

                                                   title  \
0      The Brands Queen Elizabeth, Prince Charles, an…
1                           50 Worst Habits For Belly Fat
2      The Cost of Trump's Aid Freeze in the Trenches…
3      I Was An NBA Wife. Here's How It Affected My M…
```

```
4      How to Get Rid of Skin Tags, According to a De…
…                                                    …
51277  Adapting, Learning And Soul Searching: Reflect…
51278  Family says 13-year-old Broadway star died fro…
51279  St. Dominic soccer player tries to kick cancer…
51280                       How the Sounders won MLS Cup
51281                  Best Sports Car Deals for October

                                                 abstract  \
0      Shop the notebooks, jackets, and more that the…
1      These seemingly harmless habits are holding yo…
2      Lt. Ivan Molchanets peeked over a parapet of s…
3      I felt like I was a fraud, and being an NBA wi…
4      They seem harmless, but there's a very good re…
…                                                    …
51277  Woolsey Fire Anniversary: A community is forev…
51278                                                NaN
51279  Sometimes, what happens on the sidelines can b…
51280  Mark, Jeremiah and Casey were so excited they …
51281                                                NaN

                                                   url  \
0      https://assets.msn.com/labs/mind/AAGH0ET.html
1      https://assets.msn.com/labs/mind/AAB19MK.html
2      https://assets.msn.com/labs/mind/AAJgNsz.html
3      https://assets.msn.com/labs/mind/AACk2N6.html
4      https://assets.msn.com/labs/mind/AAAKEkt.html
…                                                  …
51277  https://assets.msn.com/labs/mind/BBWzQJK.html
51278  https://assets.msn.com/labs/mind/BBWzQYV.html
51279  https://assets.msn.com/labs/mind/BBWzQnK.html
51280  https://assets.msn.com/labs/mind/BBWzQuK.html
51281  https://assets.msn.com/labs/mind/BBy5rVe.html

                                          title_entities  \
0      [{"Label": "Prince Philip, Duke of Edinburgh",…
1      [{"Label": "Adipose tissue", "Type": "C", "Wik…
2                                                     []
3                                                     []
4      [{"Label": "Skin tag", "Type": "C", "WikidataI…
…                                                    …
51277  [{"Label": "Woolsey Fire", "Type": "N", "Wikid…
51278  [{"Label": "Broadway theatre", "Type": "F", "W…
51279                                                 []
51280  [{"Label": "MLS Cup", "Type": "U", "WikidataId…
51281  [{"Label": "Peugeot RCZ", "Type": "V", "Wikida…
```

```
                                          abstract_entities
0                                                        []
1       [{"Label": "Adipose tissue", "Type": "C", "Wik…
2       [{"Label": "Ukraine", "Type": "G", "WikidataId…
3       [{"Label": "National Basketball Association", …
4       [{"Label": "Skin tag", "Type": "C", "WikidataI…
…                                                      …
51277   [{"Label": "Woolsey Fire", "Type": "N", "Wikid…
51278                                                    []
51279                                                    []
51280                                                    []
51281                                                    []

[51282 rows x 8 columns]
```

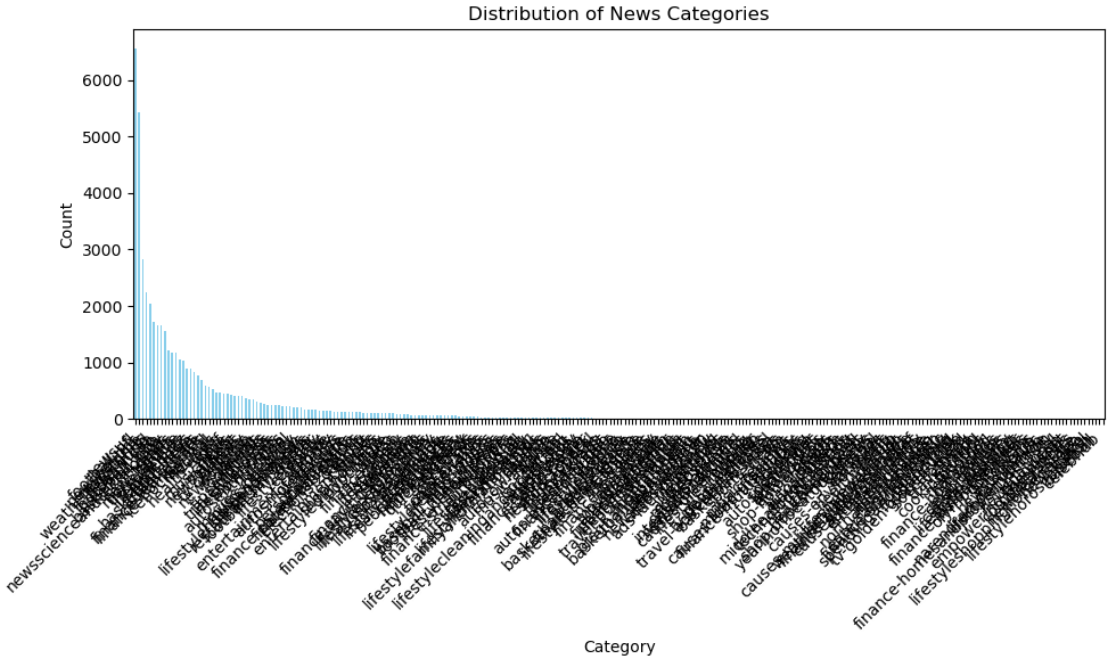
2. :

[20]:
```python
all_categories = news_df['subcategory']

#
def find_frequent_itemsets(categories, min_support=0.01):
    category_counts = categories.value_counts(normalize=True)
    frequent_categories = category_counts[category_counts >= min_support].index.
 ↪tolist()
    return frequent_categories

#
frequent_categories = find_frequent_itemsets(all_categories)

#
print("Frequent categories:")
for category in frequent_categories:
    print(category)
```

```
Frequent categories:
newsus
football_nfl
newspolitics
newscrime
weathertopstories
newsworld
football_ncaa
baseball_mlb
basketball_nba
newsscienceandtechnology
news
newstrends
```

```
more_sports
travelarticle
travelnews
lifestylebuzz
autosnews
basketball_ncaa
financenews
finance-real-estate
finance-companies
icehockey_nhl
```

[21]:
```
import matplotlib.pyplot as plt
category_counts = news_df['subcategory'].value_counts()

#
plt.figure(figsize=(10, 6))
category_counts.plot(kind='bar', color='skyblue')
plt.title('Distribution of News Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

#
plt.show()
```



3. :

Apriori FP-growth Apriori

[22]:
```
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
#      Apriori
user_data = []
last_user = '24186'
tmp = []

for index, row in news_df.iterrows():
    user_id = row['category']
    vroot_id = row['subcategory']

    if user_id == last_user:
        tmp.append(vroot_id)
    else:
        user_data.append(tmp)
        tmp = []
        tmp.append(vroot_id)
    last_user = user_id

user_data.append(tmp)
# user_data

te = TransactionEncoder()
data_encoded = te.fit_transform(user_data)
df = pd.DataFrame(data_encoded, columns=te.columns_)

df
```

[22]:

	ads-latingrammys	ads-lung-health	advice	animals	autosbuying	\
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
…	…	…	…	…	…	
41092	False	False	False	False	False	
41093	False	False	False	False	False	
41094	False	False	False	False	False	
41095	False	False	False	False	False	
41096	False	False	False	False	False	

	autoscartech	autosclassics	autoscompact	autosenthusiasts	\
0	False	False	False	False	
1	False	False	False	False	

```
2             False          False         False             False
3             False          False         False             False
4             False          False         False             False
…             …              …             …                 …
41092         False          False         False             False
41093         False          False         False             False
41094         False          False         False             False
41095         False          False         False             False
41096         False          False         False             False

       autoshybrids  …  voices  watch  weatherfullscreenmaps  \
0             False  …   False  False                  False
1             False  …   False  False                  False
2             False  …   False  False                  False
3             False  …   False  False                  False
4             False  …    True  False                  False
…             …  …     …    …                      …
41092         False  …   False  False                  False
41093         False  …   False  False                  False
41094         False  …   False  False                  False
41095         False  …   False  False                  False
41096         False  …   False  False                  False

       weathertopstories  weight-loss  weightloss  wellness  wines  wonder  \
0                  False        False       False     False  False   False
1                  False        False       False     False  False   False
2                  False        False        True     False  False   False
3                  False        False       False     False  False   False
4                  False        False       False     False  False   False
…                  …            …           …         …      …       …
41092              False        False       False     False  False   False
41093               True        False       False     False  False   False
41094              False        False       False     False  False   False
41095              False        False       False     False  False   False
41096              False        False       False     False  False   False

       yearinoffbeatgoodnews
0                      False
1                      False
2                      False
3                      False
4                      False
…                      …
41092                  False
41093                  False
41094                  False
41095                  False
```

```
41096                  False

[41097 rows x 264 columns]
```

[25]:
```
#  Apriori
frequent_itemsets = apriori(df, min_support=0.01, use_colnames=True)

#
print("Frequent Itemsets:")
# print(frequent_itemsets)
frequent_itemsets
```

```
Frequent Itemsets:
```

[25]:

	support	itemsets
0	0.006229	(animals)
1	0.002896	(autosclassics)
2	0.005548	(autosenthusiasts)
3	0.003090	(autosmotorcycles)
4	0.020099	(autosnews)
..	…	…
132	0.001436	(newscrime, newsscienceandtechnology, newsus)
133	0.001557	(newscrime, newsworld, newsus)
134	0.001922	(newspolitics, newsscienceandtechnology, newsus)
135	0.002458	(newsworld, newspolitics, newsus)
136	0.001071	(newsworld, newsscienceandtechnology, newsus)

```
[137 rows x 2 columns]
```

[26]:
```
rules = association_rules(frequent_itemsets, metric="confidence",␣
 ↪min_threshold=0.3)

#
print("\nAssociation Rules:")
# print(rules)
rules
```

```
Association Rules:
```

[26]:

	antecedents	consequents	\
0	(newsoffbeat)	(newsus)	
1	(newsscienceandtechnology)	(newsus)	
2	(baseball_mlb, football_ncaa)	(football_nfl)	
3	(newspolitics, newscrime)	(newsus)	
4	(newsscienceandtechnology, newscrime)	(newsus)	
5	(newsworld, newscrime)	(newsus)	
6	(newspolitics, newsscienceandtechnology)	(newsus)	

```
7                 (newsworld, newspolitics)        (newsus)
8     (newsworld, newsscienceandtechnology)        (newsus)

   antecedent support  consequent support   support  confidence      lift  \
0            0.009611            0.134803  0.003090    0.321519  2.385102
1            0.028469            0.134803  0.008881    0.311966  2.314234
2            0.003504            0.114047  0.001144    0.326389  2.861874
3            0.007519            0.134803  0.003285    0.436893  3.240975
4            0.003382            0.134803  0.001436    0.424460  3.148746
5            0.004648            0.134803  0.001557    0.335079  2.485690
6            0.004502            0.134803  0.001922    0.427027  3.167785
7            0.005694            0.134803  0.002458    0.431624  3.201886
8            0.002482            0.134803  0.001071    0.431373  3.200021

   leverage  conviction  zhangs_metric
0  0.001795    1.275197       0.586367
1  0.005044    1.257491       0.584533
2  0.000744    1.315229       0.652866
3  0.002271    1.536470       0.696689
4  0.000980    1.503280       0.684729
5  0.000931    1.301202       0.600488
6  0.001315    1.510014       0.687416
7  0.001690    1.522226       0.691622
8  0.000736    1.521553       0.689213
```

[27]:
```
#
rules['lift'] = rules['lift'].apply(lambda x: round(x, 2))

#
observed = rules['support'] * len(df)  #
expected = rules['antecedent support'] * rules['consequent support'] * len(df) ␣
 ↪#
chi_squared = ((observed - expected) ** 2 / expected).sum()  #

#
print("    :")
print(rules[['antecedents', 'consequents', 'lift']])
print("\n    :")
print("  :", chi_squared)
```

```
    :
                              antecedents     consequents  lift
0                          (newsoffbeat)        (newsus)  2.39
1                (newsscienceandtechnology)        (newsus)  2.31
2              (baseball_mlb, football_ncaa)  (football_nfl)  2.86
3                  (newspolitics, newscrime)        (newsus)  3.24
4     (newsscienceandtechnology, newscrime)        (newsus)  3.15
5                    (newsworld, newscrime)        (newsus)  2.49
```

```
6  (newspolitics, newsscienceandtechnology)        (newsus)  3.17
7                 (newsworld, newspolitics)        (newsus)  3.20
8    (newsworld, newsscienceandtechnology)        (newsus)  3.20
```

```
   :
  : 1120.7108790076857
```

4.

"support" "confidence" "lift"

[ ]:
```
0                            (newsoffbeat)        (newsus)  2.39
1               (newsscienceandtechnology)        (newsus)  2.31
2            (baseball_mlb, football_ncaa)  (football_nfl)  2.86
3                (newspolitics, newscrime)        (newsus)  3.24
4    (newsscienceandtechnology, newscrime)        (newsus)  3.15
5                   (newsworld, newscrime)        (newsus)  2.49
6  (newspolitics, newsscienceandtechnology)        (newsus)  3.17
7                 (newsworld, newspolitics)        (newsus)  3.20
8    (newsworld, newsscienceandtechnology)        (newsus)  3.20
```