Ashpak Sheikh

ASSOCIATION RULES

BATCH : DSWDMOD 020421

Problem Statement: -

Kitabi Duniya , a famous book store in India, which was established before Independence, the growth of the company was incremental year by year, but due to online selling of books and wide spread Internet access its annual growth started to collapse, seeing sharp downfalls, you as a Data Scientist help this heritage book store gain its popularity back and increase footfall of customers and provide ways the business can improve exponentially, apply Association Rule Algorithm, explain the rules, and visualize the graphs for clear understanding of solution.

About Data:-

We have given data about a book store containing sales of different book sales transactions containing types of books in each transaction.

Analysis with Python: -

Importing required packages to read and manipulate deta

```
import pandas as pd

import numpy as np
```

Changing display options to see entire output

```
pd. set_option('display.max_columns', None)

pd. set_option('display.max_rows', None)
```

Loading deta set

```
books=pd.read_csv("D:/DataScience/Class/assignment working/Association rule/book.csv")
```

```
In [28]: books.describe()
Out[28]:
           ChildBks     YouthBks  ...      ItalArt     Florence
count  2000.000000  2000.000000  ...  2000.000000  2000.000000
mean      0.423000     0.247500  ...     0.048500     0.108500
std       0.494159     0.431668  ...     0.214874     0.311089
min       0.000000     0.000000  ...     0.000000     0.000000
25%       0.000000     0.000000  ...     0.000000     0.000000
50%       0.000000     0.000000  ...     0.000000     0.000000
75%       1.000000     0.000000  ...     0.000000     0.000000
max       1.000000     1.000000  ...     1.000000     1.000000

[8 rows x 11 columns]

In [29]: 
```

Removing empty transactions

```
book_1=books.replace(0,np.NaN)
```

Checking Na values

```
book_1.isna().sum()
```

```
In [30]: book_1.isna().sum()
Out[30]:
ChildBks      1154
YouthBks      1505
CookBks       1138
DoItYBks      1436
RefBks        1571
ArtBks        1518
GeogBks       1448
ItalCook      1773
ItalAtlas     1926
ItalArt       1903
Florence      1783
dtype: int64
```

Droping transaction full of NA

```
book_1.dropna(how="all",inplace=True)
```

Retriving original Zeros

```
book_1.replace(np.NaN,0,inplace=True)
```

Importing package to apply association rules

```
from mlxtend.frequent_patterns import apriori,association_rules
```

Applying apriori algorithm to calculate frequent items

```
frequent_items=apriori(book_1, min_support=0.05, use_colnames=True, max_len=5)

frequent_items.sort_values("support",ascending=False,inplace=True,ignore_index=True)

frequent_items
```

Ashpak Sheikh

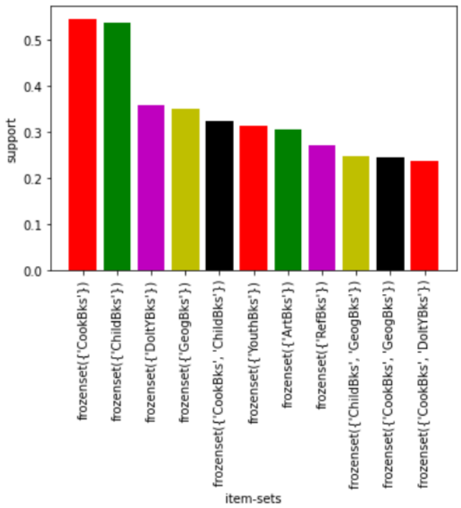
ASSOCIATION RULES

BATCH : DSWDMOD 020421

```
In [37]:
frequent_items.sort_values("support",ascending=False,inplace=True,ignore_index=True
)

In [38]: frequent_items
Out[38]:
      support                                   itemsets
0    0.545915                                  (CookBks)
1    0.535782                                 (ChildBks)
2    0.357188                                 (DoItYBks)
3    0.349588                                  (GeogBks)
4    0.324256                        (CookBks, ChildBks)
..        ...                                        ...
130  0.051932                  (ItalArt, CookBks, ArtBks)
131  0.051932                          (ItalArt, CookBks)
132  0.051298  (ChildBks, RefBks, ArtBks, GeogBks, CookBks)
133  0.050032         (GeogBks, ArtBks, DoItYBks, YouthBks)
134  0.050032           (RefBks, CookBks, ArtBks, YouthBks)

[135 rows x 2 columns]
```

visualizing frequent items

```
import matplotlib.pyplot as plt

plt.bar(x = list(range(0, 11)), height = frequent_items.support[0:11], color ='rgmyk')

plt.xticks(list(range(0, 11)), frequent_items.itemsets[0:11], rotation=90)

plt.xlabel('item-sets')

plt.ylabel('support')

plt.show()
```



Ashpak Sheikh

ASSOCIATION RULES

BATCH : DSWDMOD 020421

Applying association rules to form new rules on detaset

rules=association_rules(frequent_items,metric="lift",min_threshold=1)

#reading top 5 rules

rules.head()

Sorting rules by descending order and printing top 10 rules formed

rules.sort_values("lift",ascending=False,ignore_index=True).head(10)

```
In [44]: #sorting rules by descending order and printing top 10 rules formed

In [45]: rules.sort_values("lift",ascending=False,ignore_index=True).head(10)
Out[45]:
            antecedents             consequents  ...  leverage  conviction
0             (ItalArt)      (ArtBks, CookBks)  ...  0.038937    5.098797
1     (ArtBks, CookBks)              (ItalArt)  ...  0.038937    1.243976
2              (ArtBks)              (ItalArt)  ...  0.042679    1.175039
3              (ArtBks)     (ItalArt, CookBks)  ...  0.036079    1.142422
4    (ItalArt, CookBks)               (ArtBks)  ...  0.036079         inf
5             (ItalArt)               (ArtBks)  ...  0.042679         inf
6    (ChildBks, CookBks)     (RefBks, ItalCook)  ...  0.033467    1.123180
7     (RefBks, ItalCook)    (ChildBks, CookBks)  ...  0.033467    6.284421
8   (ChildBks, ItalCook)    (CookBks, YouthBks)  ...  0.036173    1.732271
9    (CookBks, YouthBks)   (ChildBks, ItalCook)  ...  0.036173    1.246195

[10 rows x 9 columns]
```

Insights and Summary:-

- Data given was already pre processed and was given in the form of Sparse matrix
- CookBks and ChildBks are the vary high support product with maximum frequency
- ItalArt and ArtBks,CookBks have the highest “lift” amongst all