

Product Recommendation

Import relevance library

```
In [237... import numpy as np
import pandas as pd
import glob
import matplotlib.pyplot as plt
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import seaborn as sns
import numpy as np
from sklearn.metrics import jaccard_score
```

Loaded CSV DATA

Data collected by student in class BADS 7205

```
In [238... recommend = pd.read_excel('./Customer Preference Survey.xlsx')
```

Exploratory data

Show sample data

```
In [239... recommend.head(10)
```

Out[239...]

	ประหันธ์เวลา	จักรยาน เสือ ภูเขา	คอลล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กัญชา	กัญ ขง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	...	มะม่วง เบา	ห้องน้ำ แมว อัตโนมัติ
0	2021-03-20 11:29:58.994	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1	2021-03-20 11:33:10.114	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
2	2021-03-20 11:53:44.586	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่เคย	...	ไม่เคย	ไม่เคย
3	2021-03-20 11:54:00.984	ไม่เคย	เคย	ไม่เคย	เคย	ไม่เคย	ไม่เคย	ไม่ เคย	เคย	ไม่เคย	...	เคย	ไม่เคย
4	2021-03-20 11:54:05.084	เคย	ไม่ เคย	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่เคย	ไม่ เคย	เคย	เคย	...	เคย	ไม่เคย
5	2021-03-20 11:54:12.558	ไม่เคย	ไม่ เคย	ไม่เคย	เคย	ไม่เคย	เคย	ไม่ เคย	ไม่เคย	ไม่เคย	...	ไม่เคย	ไม่เคย
6	2021-03-20 11:54:18.477	ไม่เคย	เคย	ไม่เคย	ไม่ เคย	เคย	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่เคย	...	ไม่เคย	ไม่เคย
7	2021-03-20 11:54:20.911	เคย	เคย	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่เคย	ไม่ เคย	เคย	ไม่เคย	...	เคย	ไม่เคย

	ประทับเวลา	จักรยาน เสือ ภูเขา	คอล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กัญชา	กัญ ชง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	...	มะม่วง เบา	ห้องน้ำ แมว อัตโนมัติ
8	2021-03-20 11:54:33.800	ไม่เคย	ไม่ เคย	ไม่เคย	เคย	ไม่เคย	ไม่เคย	ไม่ เคย	เคย	ไม่เคย	...	เคย	ไม่เคย
9	2021-03-20 11:54:40.338	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่ เคย	ไม่เคย	ไม่เคย	ไม่ เคย	เคย	ไม่เคย	...	ไม่เคย	ไม่เคย

10 rows × 62 columns

Data Cleansing

In [240...] `recomment.shape`

Out[240...] (48, 62)

In [241...] `recomment.columns #เครื่องย่อยเศษอาหาร`

Out[241...] Index(['ประทับเวลา', 'จักรยานเสือภูเขา', 'คอลลาเจน', 'Brownie อวกาศ', 'แม่โขง', 'Botox', 'กัญชา', 'กัญชง', 'เครื่องวัดความดัน', 'หลอดไฟเปลี่ยนสีได้', 'คอร์ส bikini wax', 'เก้าอี้นวดไฟฟ้า', 'ต้นไม้ประดับห้อง', 'หมอนข้าง', 'ตาข่าย', 'คอร์สโยท็อกกราฟี', 'แสงโสม', 'กะหรี(บับ)', 'Lego', 'แฟน ที่แปลว่าพัดลม', 'ยาตอง', 'ชุดนอนไม้ไผ่', 'ที่ยกน้ำหนัก (Dumbell)', 'เครื่องดูดฝุ่นอัตโนมัติ (robot)', 'เนือวากิล', 'หมูแดดเดียว', 'drone', 'เครื่องย่อยเศษอาหาร', 'เสื้อซีทรู', 'ไฟยิปซี', 'เนือย่าง', 'เครื่องทำลายเอกสาร', 'กระเป๋า', 'ขามู', 'Surfskate', 'Board game', 'เครื่องย่อยเศษอาหาร.1', 'ชุดน้องหมา', 'Omakase', 'เม็ดมะม่วงหิมพานต์', 'ไฟอัจฉริยะ', 'แบตเตอรี่ที่ติดผม', 'ไดร์เป่าผม Dyson', 'แผ่นเสียง', 'กรรไกรตัดขนจุก', 'PS5', 'เครื่องตรวจน้ำตาลในเลือด', 'โรตี', 'กะปิตอง', 'น้องหมา', 'แพ็คเกจไทยเที่ยวไทย', 'เกมส์ FIFA Online', 'มะม่วงเบา', 'ห้องน้ำแมวอัตโนมัติ', 'กระเป๋า chanel', 'ข้าวขาหมูคากิคาจิกอนไลน์', 'ถังเช่า', 'หมอนสุขภาพสำหรับคนนอนไม่หลับ', 'อาหารแมวคลีนเพื่อสุขภาพลดหุ่น', 'น้ำส้มดีโต้', 'Software ถูกลิขสิทธิ์', 'ครีมกันแดด'], dtype='object')

In [242...] `for i in recomment :
 if 'เครื่องย่อยเศษอาหาร' in i :
 print(i)`

เครื่องย่อยเศษอาหาร

เครื่องย่อยเศษอาหาร.1

Removed missing valued and duplicated columns value

In [243...] `recomment = recomment.drop([0,1])
missing = recomment.isnull().sum()
dd = pd.DataFrame(missing,columns = {'NUm'})
dd[dd.NUm > 0]`

Out[243...] `NUm`

	NUm
หมูแดดเดียว	1
เครื่องย่อยเศษอาหาร.1	1
เครื่องตรวจน้ำตาลในเลือด	1
แพ็คเกจไทยเที่ยวไทย	1
กระเป๋า chanel	1
ถังเช่า	1

In [244...] `recomment1 = recomment.copy()`

In [245...] `missing = recomment1.isnull().sum()
dd = pd.DataFrame(missing, columns = {'NUm'})
dd[dd.NUm > 0]`

Out[245...]

	NUm
หมูแดดเดียว	1
เครื่องย่อยเศษอาหาร.1	1
เครื่องตรวจน้ำตาลในเลือด	1
แพ็คเกจไทยเที่ยวไทย	1
กระเป๋า chanel	1
ถังเช่า	1

Encoding value & Missing value handling

Impute null values by median

In [246...] `recomment1 = recomment1.replace('เคย',1)
recomment1 = recomment1.replace('ไม่เคย',0)
recomment1['Transaction_id'] = [i for i in range(1,(recomment.shape[0]+1))]
recomment1[['หมูแดดเดียว', 'เครื่องย่อยเศษอาหาร.1', 'เครื่องตรวจน้ำตาลในเลือด', 'แพ็คเกจไทยเที่ยวไทย',
value= recomment1[['หมูแดดเดียว', 'เครื่องย่อยเศษอาหาร.1', 'เครื่องตรวจน้ำตาลในเลือด', 'แพ็คเกจไ`

In [247...] `recomment1.head(10)`

Out[247...]

	ประหันเวลา	จักรยาน เสือ ภูเขา	คอ ล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กล้วย ชา	กล้วย ขง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	...	ห้องน้ำ แมว อัตโนมัติ	กระเป๋า chanel
2	2021-03-20 11:53:44.586	0	0	0	0	0	0	0	0	0	...	0	0.0
3	2021-03-20 11:54:00.984	0	1	0	1	0	0	0	1	0	...	0	0.0

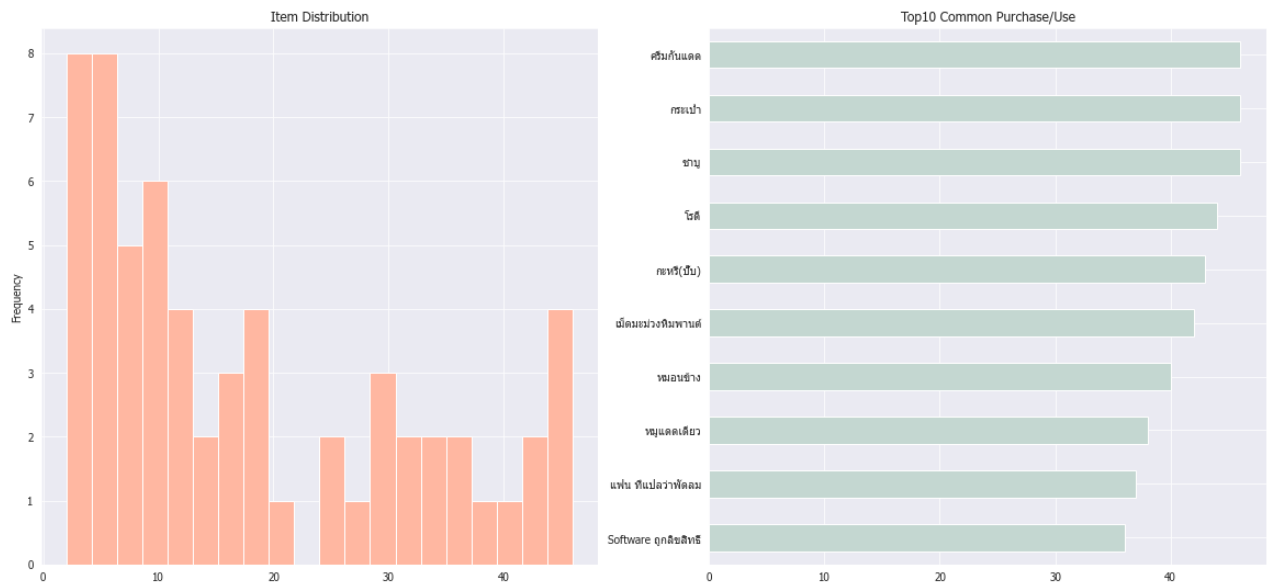
	ประทับเวลา	จักรยาน เสือ ภูเขา	คอ ล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กัญชา	กัญ ชง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	...	ห้องน้ำ แมว อัตโนมัติ	กระเป๋า chanel
4	2021-03-20 11:54:05.084	1	0	0	0	0	0	0	1	1	...	0	0.0
5	2021-03-20 11:54:12.558	0	0	0	1	0	1	0	0	0	...	0	0.0
6	2021-03-20 11:54:18.477	0	1	0	0	1	0	0	0	0	...	0	1.0
7	2021-03-20 11:54:20.911	1	1	0	0	0	0	0	1	0	...	0	0.0
8	2021-03-20 11:54:33.800	0	0	0	1	0	0	0	1	0	...	0	0.0
9	2021-03-20 11:54:40.338	0	0	0	0	0	0	0	1	0	...	0	0.0
10	2021-03-20 11:54:45.776	0	0	0	0	0	1	0	1	0	...	0	0.0
11	2021-03-20 11:54:46.611	0	1	0	0	0	0	0	1	0	...	0	0.0

10 rows × 63 columns

Distribution Item

In [248...

```
figure, axes = plt.subplots(1, 2, figsize=(20,9))
recomment1.iloc[:,1:-1].sum().plot.hist(bins= 20,color = '#ffb7a1' ,ax=axes[0],title= '
recomment1.iloc[:,1:-1].sum().sort_values(ascending= True).tail(10).plot.barh(color = '
#plt.subplot(1, 2, 1)
#plt.plot()
#plt.title('Distribution: Item')
plt.savefig('Distribution.png',format ='png',dpi = 150)
```



Removed duplicated column

In [249...

```
#เครื่องย่อยเศษอาหาร.1
recommend1 =recommend1.drop(columns=['เครื่องย่อยเศษอาหาร.1'])
```

Show over all statistics

In [250...

```
recommend1.describe().T.sort_values('mean',ascending = False)
```

Out[250...

	count	mean	std	min	25%	50%	75%	max
Transaction_id	46.0	23.500000	13.422618	1.0	12.25	23.5	34.75	46.0
ครีมกันแดด	46.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
กระเป๋	46.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
ขามู	46.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
โรตี่	46.0	0.956522	0.206185	0.0	1.00	1.0	1.00	1.0
กะหร(ปีบ)	46.0	0.934783	0.249637	0.0	1.00	1.0	1.00	1.0
เม็ดมะม่วงหิมพานต์	46.0	0.913043	0.284885	0.0	1.00	1.0	1.00	1.0
หมอนข้าง	46.0	0.869565	0.340503	0.0	1.00	1.0	1.00	1.0
หมูแดดเดียว	46.0	0.826087	0.383223	0.0	1.00	1.0	1.00	1.0
แฟน ที่แปลว่าพัดลม	46.0	0.804348	0.401085	0.0	1.00	1.0	1.00	1.0
Software ถูกลิขสิทธิ์	46.0	0.782609	0.417029	0.0	1.00	1.0	1.00	1.0
ต้นไม้ประดับห้อง	46.0	0.717391	0.455243	0.0	0.00	1.0	1.00	1.0
เนืออย่าง	46.0	0.717391	0.455243	0.0	0.00	1.0	1.00	1.0
เนือวากิล	46.0	0.673913	0.473960	0.0	0.00	1.0	1.00	1.0
ดาซัง	46.0	0.673913	0.473960	0.0	0.00	1.0	1.00	1.0
น้ำส้มดีโต้	46.0	0.652174	0.481543	0.0	0.00	1.0	1.00	1.0
Lego	46.0	0.630435	0.488021	0.0	0.00	1.0	1.00	1.0

	count	mean	std	min	25%	50%	75%	max
Board game	46.0	0.630435	0.488021	0.0	0.00	1.0	1.00	1.0
ที่ยกน้ำหนัก (Dumbell)	46.0	0.608696	0.493435	0.0	0.00	1.0	1.00	1.0
เครื่องดูดฝุ่นอัตโนมัติ (robot)	46.0	0.543478	0.503610	0.0	0.00	1.0	1.00	1.0
คอลลาเจน	46.0	0.543478	0.503610	0.0	0.00	1.0	1.00	1.0
แพ็คเกจไทยเที่ยวไทย	46.0	0.456522	0.503610	0.0	0.00	0.0	1.00	1.0
เครื่องวัดความดัน	46.0	0.413043	0.497821	0.0	0.00	0.0	1.00	1.0
กรรไกรตัดขนจมูก	46.0	0.391304	0.493435	0.0	0.00	0.0	1.00	1.0
มะม่วงเบา	46.0	0.391304	0.493435	0.0	0.00	0.0	1.00	1.0
หมอนสุขภาพสำหรับคนนอนไม่หลับ	46.0	0.391304	0.493435	0.0	0.00	0.0	1.00	1.0
ชุดน่องหมา	46.0	0.347826	0.481543	0.0	0.00	0.0	1.00	1.0
จักรยานเสือภูเขา	46.0	0.347826	0.481543	0.0	0.00	0.0	1.00	1.0
แสงโสม	46.0	0.347826	0.481543	0.0	0.00	0.0	1.00	1.0
ไฟอัจฉริยะ	46.0	0.326087	0.473960	0.0	0.00	0.0	1.00	1.0
...
ยาดอง	46.0	0.260870	0.443961	0.0	0.00	0.0	0.75	1.0
ไฟยิปซี	46.0	0.239130	0.431266	0.0	0.00	0.0	0.00	1.0
หลอดไฟเปลี่ยนสีได้	46.0	0.239130	0.431266	0.0	0.00	0.0	0.00	1.0
เสื้อขีทรู	46.0	0.239130	0.431266	0.0	0.00	0.0	0.00	1.0
น่องหมา	46.0	0.217391	0.417029	0.0	0.00	0.0	0.00	1.0
เครื่องตรวจน้ำตาลในเลือด	46.0	0.217391	0.417029	0.0	0.00	0.0	0.00	1.0
Omakase	46.0	0.217391	0.417029	0.0	0.00	0.0	0.00	1.0
Botox	46.0	0.195652	0.401085	0.0	0.00	0.0	0.00	1.0
แผ่นเสียง	46.0	0.195652	0.401085	0.0	0.00	0.0	0.00	1.0
กะปุดอง	46.0	0.195652	0.401085	0.0	0.00	0.0	0.00	1.0
แบตเตอรี่ที่ตัดผม	46.0	0.173913	0.383223	0.0	0.00	0.0	0.00	1.0
กัญชา	46.0	0.173913	0.383223	0.0	0.00	0.0	0.00	1.0
ถังเช่า	46.0	0.173913	0.383223	0.0	0.00	0.0	0.00	1.0
เครื่องทำลายเอกสาร	46.0	0.152174	0.363158	0.0	0.00	0.0	0.00	1.0
ชุดนอนไม้ไค้นอน	46.0	0.152174	0.363158	0.0	0.00	0.0	0.00	1.0
drone	46.0	0.130435	0.340503	0.0	0.00	0.0	0.00	1.0
ข้าวขานมูดาภิบาลจกออนไลน์	46.0	0.130435	0.340503	0.0	0.00	0.0	0.00	1.0
เกมส์ FIFA Online	46.0	0.130435	0.340503	0.0	0.00	0.0	0.00	1.0
เก้าอี้นวดไฟฟ้า	46.0	0.130435	0.340503	0.0	0.00	0.0	0.00	1.0

	count	mean	std	min	25%	50%	75%	max
กระเป๋ា chanel	46.0	0.108696	0.314696	0.0	0.00	0.0	0.00	1.0
ไตร์เป่าผม Dyson	46.0	0.108696	0.314696	0.0	0.00	0.0	0.00	1.0
Brownie อวภาค	46.0	0.108696	0.314696	0.0	0.00	0.0	0.00	1.0
คอร์สโบท็อกแกรม	46.0	0.108696	0.314696	0.0	0.00	0.0	0.00	1.0
คอร์ส bikini wax	46.0	0.086957	0.284885	0.0	0.00	0.0	0.00	1.0
กัญชง	46.0	0.086957	0.284885	0.0	0.00	0.0	0.00	1.0
อาหารแมวคลีนเพื่อสุขภาพลดหุ่น	46.0	0.086957	0.284885	0.0	0.00	0.0	0.00	1.0
ห้องน้ำแมวอัดโนมัติ	46.0	0.065217	0.249637	0.0	0.00	0.0	0.00	1.0
Surfskate	46.0	0.065217	0.249637	0.0	0.00	0.0	0.00	1.0
PS5	46.0	0.043478	0.206185	0.0	0.00	0.0	0.00	1.0
เครื่องย่อยเศษอาหาร	46.0	0.043478	0.206185	0.0	0.00	0.0	0.00	1.0

61 rows × 8 columns

In [251...

recomment1.iloc[:,1].value_counts()

Out[251...

0 30
1 16
Name: จักรยานเสือภูเขา, dtype: int64

In [255...

recomment1.head(10)

Out[255...

	ประหับเวลา	จักรยาน เสือ ภูเขา	คอ ล ลา เจน	Brownie อวภาค	แม่ โขง	Botox	กัญชา	กัญ ชง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	...	ห้องน้ำ แมว อัดโนมัติ	กระเป๋า chanel
2	2021-03-20 11:53:44.586	0	0	0	0	0	0	0	0	0	...	0	0.0
3	2021-03-20 11:54:00.984	0	1	0	1	0	0	0	1	0	...	0	0.0
4	2021-03-20 11:54:05.084	1	0	0	0	0	0	0	1	1	...	0	0.0
5	2021-03-20 11:54:12.558	0	0	0	1	0	1	0	0	0	...	0	0.0
6	2021-03-20 11:54:18.477	0	1	0	0	1	0	0	0	0	...	0	1.0
7	2021-03-20 11:54:20.911	1	1	0	0	0	0	0	1	0	...	0	0.0
8	2021-03-20 11:54:33.800	0	0	0	1	0	0	0	1	0	...	0	0.0
9	2021-03-20 11:54:40.338	0	0	0	0	0	0	0	1	0	...	0	0.0

	ประทับเวลา	จักรยาน เสือ ภูเขา	คอล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กัญชา	กัญ ขง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	...	ห้องน้ำ แมว อัตโนมัติ	กระเป๋ chanel
10	2021-03-20 11:54:45.776	0	0	0	0	0	1	0	1	0	...	0	0.0
11	2021-03-20 11:54:46.611	0	1	0	0	0	0	0	1	0	...	0	0.0

10 rows × 62 columns

Market Basket Analysis

Association rule & Frequent Items set

find whole support score show the association of occurrent items

```
In [256...] frequent_itemsets = apriori(recomment1.iloc[:,1:-1], min_support=0.5, use_colnames=True)
print(frequent_itemsets.shape)
frequent_itemsets.head(10)
```

(5871, 2)

```
Out[256...] support itemsets
0 0.543478 (คอลลาเจน)
1 0.717391 (ต้นไม้ประดับห้อง)
2 0.869565 (หมอนข้าง)
3 0.673913 (ตาข่าย)
4 0.934783 (กะหรี่ปั๊บ)
5 0.630435 (Lego)
6 0.804348 (แฟน ที่แปลว่าพัดลม)
7 0.608696 (ที่ยกน้ำหนัก (Dumbell))
8 0.543478 (เครื่องดูดฝุ่นอัตโนมัติ (robot))
9 0.673913 (เนื้อวากิว)
```

```
In [257...] # eec = List(frequent_itemsets)
# print(eec[0])
#apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2)
frequent_itemsets['item_len'] = frequent_itemsets['itemsets'].apply(lambda x : len(x))#
frequent_itemsets = frequent_itemsets.sort_values(by = ['support'],ascending = False)
frequent_itemsets.head(10)
```

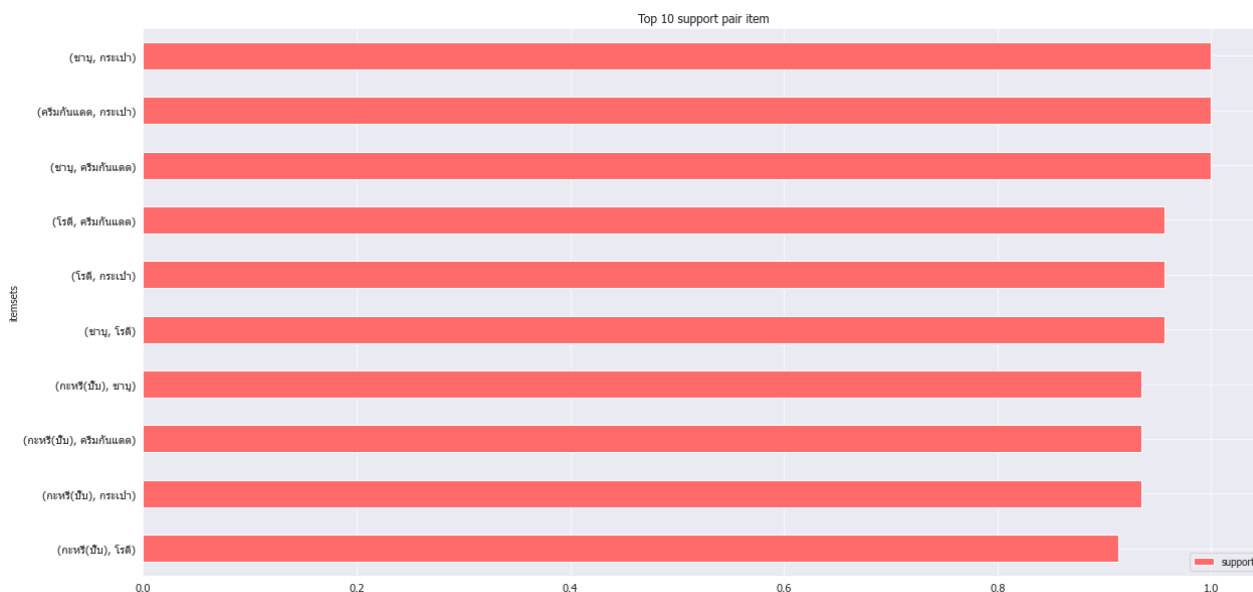
```
Out[257...] support itemsets item_len
```


	support	itemsets	item_len
12	1.000000	(กระเป๋)	1
137	1.000000	(ขานู, กระเป๋)	2
631	1.000000	(ขานู, ครีมกันแดด, กระเป๋)	3
143	1.000000	(ครีมกันแดด, กระเป๋)	2
19	1.000000	(ครีมกันแดด)	1
149	1.000000	(ขานู, ครีมกันแดด)	2
13	1.000000	(ขานู)	1
160	0.956522	(โรดี, ครีมกันแดด)	2
642	0.956522	(โรดี, ครีมกันแดด, กระเป๋)	3
140	0.956522	(โรดี, กระเป๋)	2

Show top 10 frequent items set 2 items by support score

In [258]...

```
import matplotlib.font_manager as fm
import seaborn as sn
#plt.subplots(figsize=(20,18))
sn.set_style("darkgrid",{"font.sans-serif":["simhei", 'Tahoma']})
frequent_itemsets[(frequent_itemsets['item_len'] == 2)
                   ].sort_values(by=['support'], ascending= True).tail(10).plot.barh(
                        x='itemsets',y='support',title = 'Top 10 support pair item' ,color
,figsize=(20,10))
plt.savefig('Top10_support_len2.png',format = 'png',dpi = 150)
```



In [259]...

```
frequent_2len = frequent_itemsets[frequent_itemsets['item_len'] == 2 ]
frequent_2len.head(10)
```

Out[259]...

	support	itemsets	item_len
137	1.000000	(ขานู, กระเป๋)	2

	support	itemsets	item_len
143	1.000000	(ครีมกันแดด, กระเป๋า)	2
149	1.000000	(ชาบู, ครีมกันแดด)	2
160	0.956522	(โรตีส, ครีมกันแดด)	2
140	0.956522	(โรตีส, กระเป๋า)	2
146	0.956522	(ชาบู, โรตีส)	2
75	0.934783	(กะหรี่ปั๊บ, ชาบู)	2
81	0.934783	(กะหรี่ปั๊บ, ครีมกันแดด)	2
74	0.934783	(กะหรี่ปั๊บ, กระเป๋า)	2
78	0.913043	(กะหรี่ปั๊บ, โรตีส)	2

Show top 10 frequent item set 3 items by support score

```
In [260...] frequent_3len = frequent_itemsets[frequent_itemsets['item_len'] == 3 ]
frequent_3len.sort_values('support', ascending = False).head(10) #0.956522
```

```
Out[260...]

```

	support	itemsets	item_len
631	1.000000	(ชาบู, ครีมกันแดด, กระเป๋า)	3
656	0.956522	(ชาบู, ครีมกันแดด, โรตีส)	3
628	0.956522	(ชาบู, กระเป๋า, โรตีส)	3
642	0.956522	(โรตีส, ครีมกันแดด, กระเป๋า)	3
428	0.934783	(กะหรี่ปั๊บ, กระเป๋า, ชาบู)	3
434	0.934783	(กะหรี่ปั๊บ, ครีมกันแดด, กระเป๋า)	3
440	0.934783	(กะหรี่ปั๊บ, ครีมกันแดด, ชาบู)	3
431	0.913043	(กะหรี่ปั๊บ, กระเป๋า, โรตีส)	3
627	0.913043	(ชาบู, กระเป๋า, เม็ดมะม่วงหิมพานต์)	3
437	0.913043	(กะหรี่ปั๊บ, ชาบู, โรตีส)	3

```
In [231...] ## Plot Top 10 Support Score with 3 itmes pair
sn.set_style("darkgrid",{"font.sans-serif":["simhei", 'Tahoma']})
frequent_itemsets[(frequent_itemsets['item_len'] == 3) &
                  ((frequent_itemsets['support'] > 0.9)
                   )].sort_values(by=['support'], ascending= True).tail(10).plot.barh(
                    x='itemsets',y='support',title = 'Top 10 support 3 pair items',colo
,figsize=(20,10))
plt.savefig('Top10_support.png',format = 'png',dpi = 150)
```



```
In [79]: # for item in frequent_itemsets:

#         # first index of the inner list
#         # Contains base item and add item
#         pair = item[0]
#         print(pair)
#         items = [x for x in pair]
#         print("Rule: " + items[0] + " -> " + items[1])

#         #second index of the inner list
#         print("Support: " + str(item[1]))

#         #third index of the list located at 0th
#         #of the third index of the inner list

#         print("Confidence: " + str(item[2][0][2]))
#         print("Lift: " + str(item[2][0][3]))
#         print("=====")
```

```
In [261...] frequent_2len.head()
```

```
Out[261...]
   support  itemsets  item_len
137  1.000000  (ชาบู, กระป๋อง)      2
143  1.000000  (ครีมกันแดด, กระป๋อง)      2
149  1.000000  (ชาบู, ครีมกันแดด)      2
160  0.956522  (โรตีส, ครีมกันแดด)      2
140  0.956522  (โรตีส, กระป๋อง)      2
```

Running a market basket analysis. Here's the visualization of 1-itemset association rules (filtered by Support > 0.5 and Lift > 1.01) Nodes represent items and directed edges represent rules (antecedents → consequents). Edge labels annotates Lift values. Note that for 1-to-1-itemset, both directions of rules ($A \rightarrow B$, $B \rightarrow A$) have the same lift values, so in the case, we always have bi-directional edges ($A \leftrightarrow B$).

```
In [262...] rule_sim = association_rules(frequent_itemsets, metric="lift", min_threshold= 1.0)
```

```
rule_sim = rule_sim.sort_values('lift',ascending = False)
print(rule_sim.shape)
rule_sim.head(10)
```

(383348, 9)

Out[262...

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
307195	(หมูแดดเดียว, เนื้อย่าง, ครีม กันแดด, Software ...	(ชาบู, เนื้อวากิล, เม็ด มะม่วงหิมพานต์)	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
261467	(หมูแดดเดียว, โรตีส, เนื้อย่าง, Software ถูก กลิช...	(ครีมกันแดด, เนื้อวากิล, กระเป่า, เม็ดมะ ม่วงหิ...	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
269149	(หมูแดดเดียว, โรตีส, เนื้อย่าง, Software ถูก กลิช...	(ชาบู, ครีม กันแดด, เนื้อวากิล, เม็ดมะม่วงหิม พา...	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
308700	(หมูแดดเดียว, เนื้อย่าง, กระเป่า, Software ถูก...	(โรตีส, เนื้อวากิล, เม็ด มะม่วงหิมพานต์)	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
308706	(หมูแดดเดียว, โรตีส, เนื้อย่าง, Software ถูก กลิช...	(เนื้อวากิล, กระเป่า, เม็ด มะม่วงหิมพานต์)	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
308734	(หมูแดดเดียว, เนื้อย่าง, Software ถูก ลิขสิทธิ์)	(โรตีส, เนื้อวากิล, กระเป่า, เม็ด มะม่วงหิมพานต์)	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
269143	(หมูแดดเดียว, เนื้อย่าง, ครีม กันแดด, Software ...	(ชาบู, เนื้อวากิล, โรตีส, เม็ด มะม่วงหิมพานต์)	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
261400	(Software ถูก ลิขสิทธิ์, ครีม กันแดด, กระเป่า, ...	(โรตีส, เนื้อวากิล, เม็ด มะม่วงหิมพานต์)	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
253182	(หมูแดดเดียว, เนื้อย่าง, Software ถูก ลิขสิทธิ์)	(ชาบู, เม็ด มะม่วงหิมพานต์, ครีมกันแดด, กระเป่า...	0.521739	0.652174	0.5	0.958333	1.469444	0.159735
247434	(หมูแดดเดียว, โรตีส, เนื้อย่าง, Software ถูก กลิช...	(ชาบู, เม็ด มะม่วงหิมพานต์, ครีมกันแดด, กระเป่า...	0.521739	0.652174	0.5	0.958333	1.469444	0.159735

In [300...

```
simple_rules = rules[(rules['antecedents'].apply(len) == 1) & (rules['consequents'].app
```

```
print(simple_rules.shape)

simple_rules.sort_values('lift', ascending=False).head(10)
```

(46, 9)

Out[300...

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
6383	{เนื้วากิล}	{เนื้อย่าง}	0.673913	0.717391	0.652174	0.967742	1.349	0.168715	8.7
6382	{เนื้อย่าง}	{เนื้วากิล}	0.717391	0.673913	0.652174	0.909091	1.349	0.168715	3.1
106627	{Lego}	{ตันไม้ประดับ ห้อง}	0.630435	0.717391	0.543478	0.862069	1.202	0.091210	2.0
106626	{ตันไม้ประดับ ห้อง}	{Lego}	0.717391	0.630435	0.543478	0.757576	1.202	0.091210	1.1
160753	{Lego}	{เนื้อย่าง}	0.630435	0.717391	0.521739	0.827586	1.154	0.069471	1.0
160752	{เนื้อย่าง}	{Lego}	0.717391	0.630435	0.521739	0.727273	1.154	0.069471	1.3
28348	{Software ถูก ลิขสิทธิ์}	{เนื้วากิล}	0.782609	0.673913	0.586957	0.750000	1.113	0.059546	1.3
28349	{เนื้วากิล}	{Software ถูก ลิขสิทธิ์}	0.673913	0.782609	0.586957	0.870968	1.113	0.059546	1.0
60521	{ตันไม้ประดับ ห้อง}	{เนื้อย่าง}	0.717391	0.717391	0.565217	0.787879	1.098	0.050567	1.3
60520	{เนื้อย่าง}	{ตันไม้ประดับ ห้อง}	0.717391	0.717391	0.565217	0.787879	1.098	0.050567	1.3

In [281...

```
rules_1 = rule_sim[(rule_sim.antecedents.apply(len)==1) & (rule_sim.consequents.apply(len)==1)]
print(rules_1.shape)
rules_1['antecedents'] = rules_1.antecedents.apply(lambda x : list(x)[0] )
rules_1['consequents'] = rules_1.consequents.apply(lambda x : list(x)[0] )
rules_1['lift'] = rules_1.lift.apply(lambda x : round(x,3))
# rules_1
rules_1.head(10)
```

(46, 9)

Out[281...

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
16646	เนื้อย่าง	เนื้วากิล	0.717391	0.673913	0.652174	0.909091	1.349	0.168715	3.1
16647	เนื้วากิล	เนื้อย่าง	0.673913	0.717391	0.652174	0.967742	1.349	0.168715	8.7
151190	ตันไม้ประดับ ห้อง	Lego	0.717391	0.630435	0.543478	0.757576	1.202	0.091210	1.1
151191	Lego	ตันไม้ประดับ ห้อง	0.630435	0.717391	0.543478	0.862069	1.202	0.091210	2.0
216199	Lego	เนื้อย่าง	0.630435	0.717391	0.521739	0.827586	1.154	0.069471	1.0
216198	เนื้อย่าง	Lego	0.717391	0.630435	0.521739	0.727273	1.154	0.069471	1.3
49455	เนื้วากิล	Software ถูก ลิขสิทธิ์	0.673913	0.782609	0.586957	0.870968	1.113	0.059546	1.0

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
49454	Software ถูก ลิขสิทธิ์	เนื้อวากิล	0.782609	0.673913	0.586957	0.750000	1.113	0.059546	1.1
92862	เนื้ออย่าง	ตันไม้ประดับ ห้อง	0.717391	0.717391	0.565217	0.787879	1.098	0.050567	1.1
92863	ตันไม้ประดับ ห้อง	เนื้ออย่าง	0.717391	0.717391	0.565217	0.787879	1.098	0.050567	1.1

In [277...

```
rules = association_rules(frequent_itemsets , metric="lift", min_threshold= 1.05)
rules['antecedents'] = rules.antecedents.apply(lambda x : set(x))
rules['consequents'] = rules.consequents.apply(lambda x : set(x))
rules['lift'] = rules.lift.apply(lambda x : round(x,3))
rules[ (rules.antecedents.apply(len)==2) & (rules.consequents.apply(len)==1)].sort_valu
```

Out[277...

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
28350	{หมูแดดเดียว, เนื้ออย่าง}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	4.7%
49226	{เนื้ออย่าง, Software ถูก ลิขสิทธิ์}	{เนื้อวากิล}	0.608696	0.673913	0.565217	0.928571	1.378	0.155009	4.5%
7023	{โรตีส, เนื้อวากิล}	{เนื้ออย่าง}	0.673913	0.717391	0.652174	0.967742	1.349	0.168715	8.7%
6042	{เนื้ออย่าง, กระเป๋}	{เนื้อวากิล}	0.717391	0.673913	0.652174	0.909091	1.349	0.168715	3.5%
6043	{กระเป๋, เนื้อวากิล}	{เนื้ออย่าง}	0.673913	0.717391	0.652174	0.967742	1.349	0.168715	8.7%
6876	{ขานู, เนื้ออย่าง}	{เนื้อวากิล}	0.717391	0.673913	0.652174	0.909091	1.349	0.168715	3.5%
6877	{ขานู, เนื้อวากิล}	{เนื้ออย่าง}	0.673913	0.717391	0.652174	0.967742	1.349	0.168715	8.7%
7022	{โรตีส, เนื้ออย่าง}	{เนื้อวากิล}	0.717391	0.673913	0.652174	0.909091	1.349	0.168715	3.5%
5951	{ครีมกันแดด, เนื้อวากิล}	{เนื้ออย่าง}	0.673913	0.717391	0.652174	0.967742	1.349	0.168715	8.7%
5950	{เนื้ออย่าง, ครีมกันแดด}	{เนื้อวากิล}	0.717391	0.673913	0.652174	0.909091	1.349	0.168715	3.5%

In [278...

```
rules = association_rules(frequent_itemsets , metric="lift", min_threshold= 1.05)
rules['antecedents'] = rules.antecedents.apply(lambda x : set(x))
rules['consequents'] = rules.consequents.apply(lambda x : set(x))
rules['lift'] = rules.lift.apply(lambda x : round(x,3))
rules[ (rules.antecedents.apply(len)==3) & (rules.consequents.apply(len)==1)].sort_valu
```

Out[278...

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	cc
304382	{หมูแดดเดียว, เนื้อย่าง, Software ถูก ลิขสิทธิ์}	{เนื้อวากิล}	0.521739	0.673913	0.500000	0.958333	1.422	0.148393	
37363	{หมูแดดเดียว, เนื้อย่าง, เม็ดมะม่วงหิมพานต์}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	
37351	{โรตีสี, หมูแดดเดียว, เนื้อย่าง}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	
37377	{ชาบู, หมูแดดเดียว, เนื้อย่าง}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	
37586	{หมูแดดเดียว, เนื้อย่าง, 크림ก้นแดด}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	
37388	{หมูแดดเดียว, เนื้อย่าง, กระเป๋}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	
46804	{ชาบู, เนื้อย่าง, Software ถูก ลิขสิทธิ์}	{เนื้อวากิล}	0.608696	0.673913	0.565217	0.928571	1.378	0.155009	
54362	{เนื้อย่าง, กระเป๋, Software ถูก ลิขสิทธิ์}	{เนื้อวากิล}	0.608696	0.673913	0.565217	0.928571	1.378	0.155009	
44749	{กะหรี่ปั๊, หมูแดดเดียว, เนื้อย่าง}	{เนื้อวากิล}	0.608696	0.673913	0.565217	0.928571	1.378	0.155009	
46828	{เนื้อย่าง, 크림ก้นแดด, Software ถูก ลิขสิทธิ์}	{เนื้อวากิล}	0.608696	0.673913	0.565217	0.928571	1.378	0.155009	

In [279...

```
rules = association_rules(frequent_itemsets , metric="lift", min_threshold= 1.05)
rules['antecedents'] = rules.antecedents.apply(lambda x : set(x))
rules['consequents'] = rules.consequents.apply(lambda x : set(x))
rules['lift'] = rules.lift.apply(lambda x : round(x,3))
rules[ (rules.antecedents.apply(len)==4) & (rules.consequents.apply(len)==1)].sort_valu
```

Out[279...

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	con
275683	{หมูแดดเดียว, ชาบู, เนื้อย่าง, Software ถูก ลิขสิทธิ์}	{เนื้อวากิล}	0.521739	0.673913	0.500000	0.958333	1.422	0.148393	7.1

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
275727	{เนื้ออย่าง, หมูแดดเดียว, กระเป๋, Software ถูก...	{เนื้อวากิล}	0.521739	0.673913	0.500000	0.958333	1.422	0.148393	7.1
275063	{หมูแดดเดียว, เนื้ออย่าง, ครีมกันแดด, Software ...	{เนื้อวากิล}	0.521739	0.673913	0.500000	0.958333	1.422	0.148393	7.1
275649	{เนื้ออย่าง, หมูแดดเดียว, Software ถูก ลิขสิทธิ์...	{เนื้อวากิล}	0.521739	0.673913	0.500000	0.958333	1.422	0.148393	7.1
275107	{หมูแดดเดียว, โรตี่, เนื้ออย่าง, Software ถูก ลิขสิทธิ์...	{เนื้อวากิล}	0.521739	0.673913	0.500000	0.958333	1.422	0.148393	7.1
32517	{หมูแดดเดียว, ขาหมู, เนื้ออย่าง, กระเป๋}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	4.1
32151	{หมูแดดเดียว, โรตี่, เนื้ออย่าง, เม็ดมะม่วงหิมพานต์}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	4.1
29991	{หมูแดดเดียว, เนื้ออย่าง, ครีมกันแดด, เม็ดมะม่วง...	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	4.1
32179	{หมูแดดเดียว, ขาหมู, ครีมกันแดด, เนื้ออย่าง}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	4.1
32203	{หมูแดดเดียว, เนื้ออย่าง, ครีมกันแดด, โรตี่}	{เนื้อวากิล}	0.630435	0.673913	0.586957	0.931034	1.382	0.162098	4.1

Highest lift score

In [282... `max(rules_1.lift)`

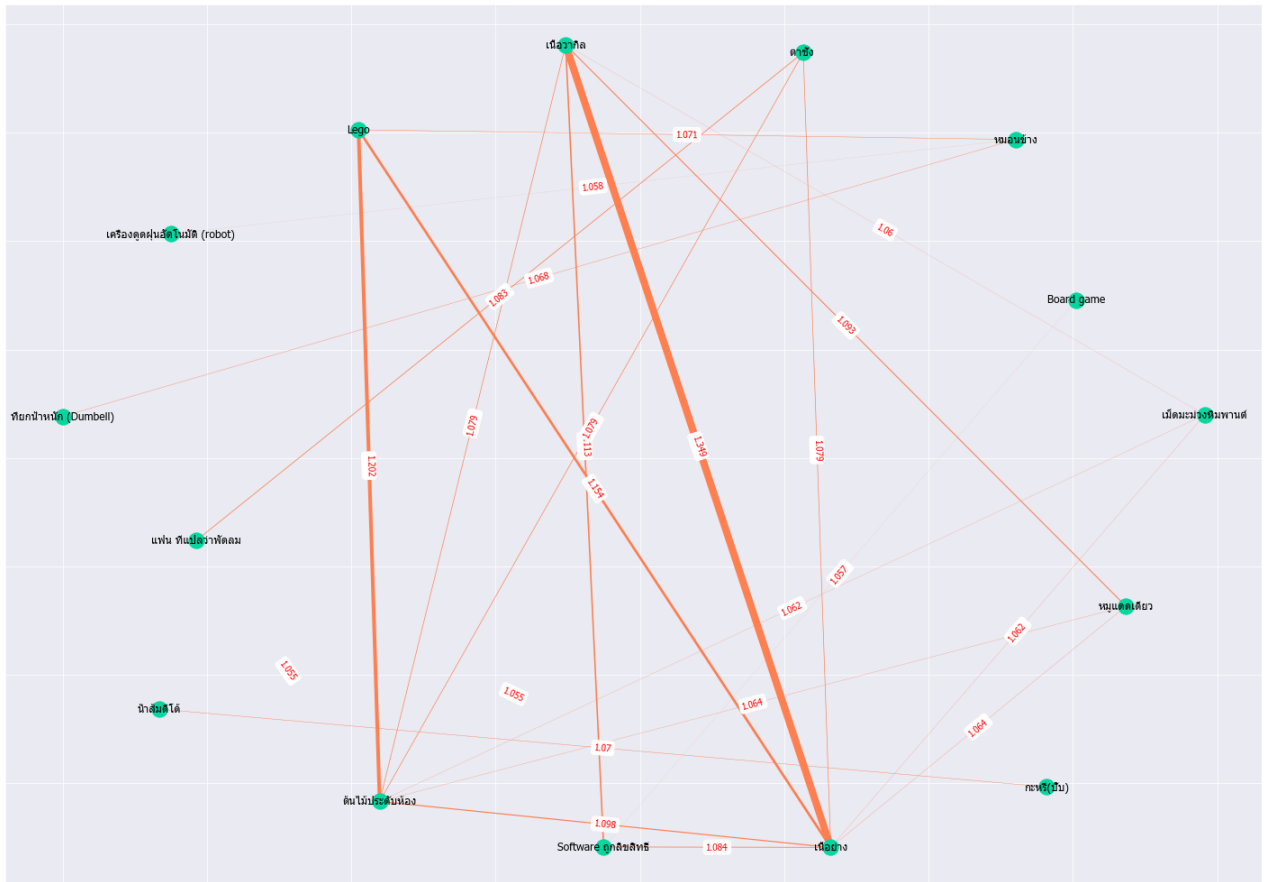
Out[282... 1.349

```
In [299... from matplotlib.pyplot import figure, text
import networkx as nx

fig, ax =plt.subplots(figsize=(25,18))
GA=nx.from_pandas_edgelist(rules_1,source='antecedents',target='consequents',edge_attr=
#GA.nodes()
```



```
# nx.draw(GA,with_labels=True,node_color='orange')
# plt.show()
weight = [ 0.5+float(i['lift'])*4 for i in dict(GA.edges).values()]
weight = ((np.array(weight)- min(weight))/(max(weight)-min(weight)))*8
labels = [i for i in dict(GA.nodes)]
labels = {i:i for i in dict(GA.nodes).keys()}
#pos = nx.spring_layout(GA)
pos = nx.spring_layout(GA,weight='weight', k=5)
nx.draw_networkx_nodes(GA, pos,ax = ax,node_color = '#06D6A0') # #ff7f50 '#06D6A0
nx.draw_networkx_edges(GA, pos,edge_color='#ff7f50',arrowsize=10, width=weight, ax=ax )
edge_labels = nx.get_edge_attributes(GA, 'lift')
#edge_labels = nx.get_edge_attributes(G, 'lift')
nx.draw_networkx_edge_labels(GA, pos, edge_labels=edge_labels, font_color='red')#, **l
_ = nx.draw_networkx_labels(GA, pos, labels, ax=ax)
plt.savefig('graph_item_1_1.png',format = 'png',dpi = 150)
```

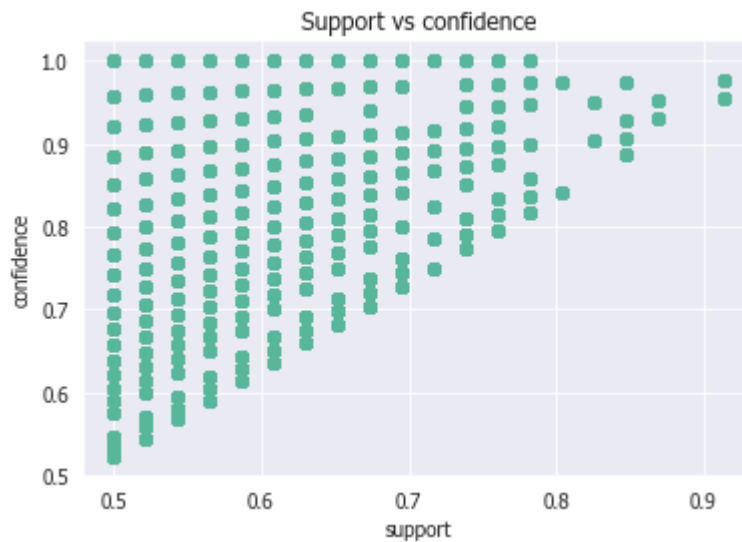


Visualizing results

Support vs Confidence

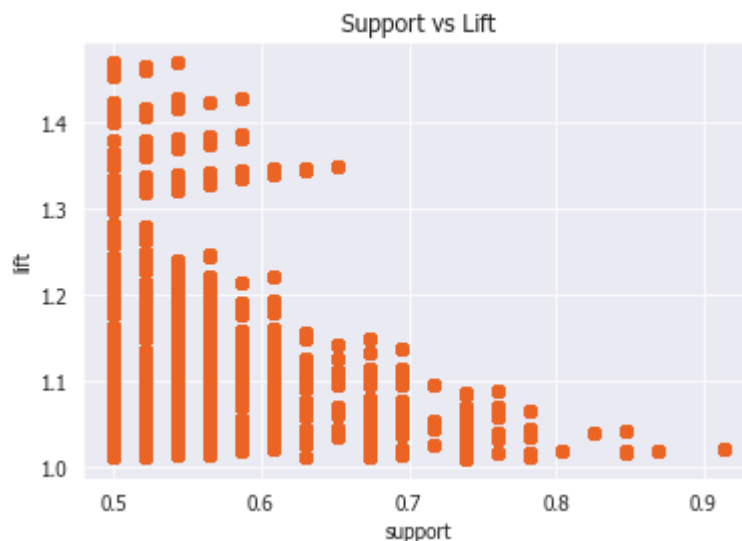
```
In [87]: rules = rules.sort_values(['lift'], ascending = False)
#display(rules)

plt.scatter(rules['support'], rules['confidence'], alpha=0.5, color = '#58b69b')
plt.xlabel('support')
plt.ylabel('confidence')
plt.title('Support vs confidence')
plt.plot(figsize=(50,50))
plt.savefig('Support vs confidence.png', format = 'png', dpi = 150)
```



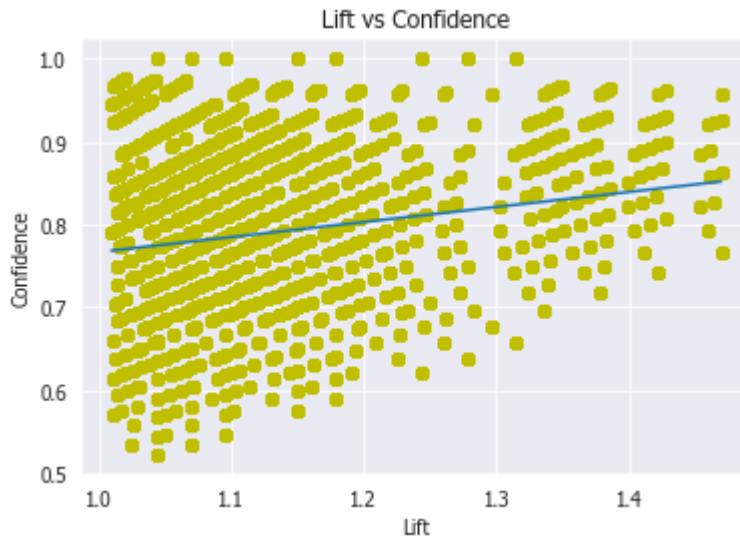
Support vs lift

```
In [88]: rules = rules.sort_values(['lift'],ascending = False)
#display(rules)
plt.scatter(rules['support'], rules['lift'], alpha=0.5, color = '#ed6423')
plt.xlabel('support')
plt.ylabel('lift')
plt.title('Support vs Lift')
plt.plot(figsize=(50,50))
plt.savefig('Support_vs_lift.png',format = 'png',dpi = 150)
```



Lift vs Confidence

```
In [89]: fit = np.polyfit(rules['lift'], rules['confidence'], 1)
fit_fn = np.poly1d(fit)
plt.plot(rules['lift'], rules['confidence'], 'yo', rules['lift'], fit_fn(rules['lift']))
plt.xlabel('Lift')
plt.ylabel('Confidence')
plt.title('Lift vs Confidence')
plt.plot(figsize=(30,30))
plt.savefig('lift_VS_confident.png',format = 'png',dpi = 150)
```



```
In [90]: fit_fn(rules['lift'])
```

```
Out[90]: array([0.85191353, 0.85191353, 0.85191353, ..., 0.7685634 , 0.7685634 ,
0.7685634 ])
```

Collaborative Filtering - Item Similarity

By using users' ratings for each item as its feature vector, calculates cosine similarity values for each pair of items. We could recommend items based on item similarity. Filtering the similarity using 0.65 threshold, and visualize them in graph. A node represents an item. Edge thickness and color represent cosine similarity between 2 items.

```
In [91]: item_item_matrix = pd.DataFrame(index=recommend1.columns,columns=recommend1.columns)
item_item_matrix1 = item_item_matrix.iloc[1:-1,1:-1]
```

```
In [92]: recommend2 = recommend1.iloc[:,1:-1]
recommend2.head(10)
```

```
Out[92]:
```

	จักรยาน เสือ ภูเขา	คอล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กัญชา	กัญ ชง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	คอร์ส bikini wax	...	มะม่วง เบา	ห้องน้ำ แมว ฮัตโนมดี	กระเป๋ chan
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0
3	0	1	0	1	0	0	0	1	0	0	...	1	0	0
4	1	0	0	0	0	0	0	1	1	0	...	1	0	0
5	0	0	0	1	0	1	0	0	0	0	...	0	0	0
6	0	1	0	0	1	0	0	0	0	0	...	0	0	1
7	1	1	0	0	0	0	0	1	0	0	...	1	0	0
8	0	0	0	1	0	0	0	1	0	0	...	1	0	0
9	0	0	0	0	0	0	0	1	0	0	...	0	0	0
10	0	0	0	0	0	1	0	1	0	0	...	0	0	0

	จักรยาน เสือ ภูเขา	คอล ลา เจน	Brownie อวกาศ	แม่ โขง	Botox	กัญชา	กัญ ชง	เครื่อง วัด ความ ดัน	หลอด ไฟ เปลี่ยน สีได้	คอร์ส bikini wax	...	มะม่วง เบา	ห้องน้ำ แมว อัดโน้มนัด	กระเป๋ chan
11	0	1	0	0	0	0	0	1	0	0	...	1	0	0

10 rows × 60 columns

Create matrix finding similarity

Matrix showing a similarity item by cosine similarity score with the item that has item rating on each user. the highest similarity scores are 1 when items A and B rating by the whole user .Otherwise similarity score has decreased by cosine distance .

```
In [93]: from scipy.spatial.distance import cosine

for i in range(0,len(item_item_matrix1.columns)) :
    # Loop through the columns for each column
    for j in range(0,len(item_item_matrix1.columns)) :
        # Fill in placeholder with cosine similarities
        item_item_matrix1.iloc[i,j] = 1-cosine(recomment2.iloc[:,i],recomment2.iloc[:,j])
```

```
In [94]: item_item_matrix1.head(10)
```

Out[94]:

	จักรยาน เสือภูเขา	คอลลา เจน	Brownie อวกาศ	แม่โขง	Botox	กัญชา	กัญชง	เครื่องวัด ความดัน	หลอดไฟ เปลี่ยนสี ได้	คอร์ส bikini wax
จักรยาน เสือภูเขา	1	0.55	0.33541	0.400892	0.416667	0.265165	0.25	0.458831	0.452267	0.
คอลลา เจน	0.55	1	0.268328	0.320713	0.533333	0.141421	0.2	0.458831	0.361814	
Brownie อวกาศ	0.33541	0.268328	1	0.478091	0.447214	0.632456	0.67082	0.102598	0.13484	0.223
แม่โขง	0.400892	0.320713	0.478091	1	0.267261	0.566947	0.400892	0.367884	0.241747	0.133
Botox	0.416667	0.533333	0.447214	0.267261	1	0.235702	0.333333	0.229416	0.301511	0.166
กัญชา	0.265165	0.141421	0.632456	0.566947	0.235702	1	0.53033	0.324443	0	
กัญชง	0.25	0.2	0.67082	0.400892	0.333333	0.53033	1	0.114708	0.150756	
เครื่อง วัดความ ดัน	0.458831	0.458831	0.102598	0.367884	0.229416	0.324443	0.114708	1	0.415029	0.114
หลอด ไฟ เปลี่ยนสี ได้	0.452267	0.361814	0.13484	0.241747	0.301511	0	0.150756	0.415029	1	0.301

	จักรยาน เสือภูเขา	คอลลา เจน	Brownie อวกาศ	แม่โขง	Botox	กัญชา	กัญชง	เครื่องวัด ความดัน	หลอดไฟ เปลี่ยนสี ได้	คิ bil v
คอร์ส bikini wax	0.125	0.4	0.223607	0.133631	0.166667	0	0	0.114708	0.301511	

10 rows × 60 columns

```
In [95]: item_item_matrix1 = item_item_matrix1.dropna()
#sns.heatmap(item_item_matrix1)
#item_item_matrix1[item_item_matrix1.isnull()]
#sns.heatmap(item_item_matrix1.dropna())
# xx = item_item_matrix1[item_item_matrix1.isnull()]
# #xx[xx.notnull()]
# xx.head()
```

```
In [96]: links = item_item_matrix.rename_axis('related item',
axis='columns').stack().reset_index()
links.columns = ['item', 'related item', 'value']
```

```
In [97]: links['value'] = links.value.apply( lambda x : round(x,3))
links.head(10)
```

```
Out[97]:
```

	item	related item	value
0	จักรยานเสือภูเขา	จักรยานเสือภูเขา	1.000
1	จักรยานเสือภูเขา	คอลลาเจน	0.550
2	จักรยานเสือภูเขา	Brownie อวกาศ	0.335
3	จักรยานเสือภูเขา	แม่โขง	0.401
4	จักรยานเสือภูเขา	Botox	0.417
5	จักรยานเสือภูเขา	กัญชา	0.265
6	จักรยานเสือภูเขา	กัญชง	0.250
7	จักรยานเสือภูเขา	เครื่องวัดความดัน	0.459
8	จักรยานเสือภูเขา	หลอดไฟเปลี่ยนสีได้	0.452
9	จักรยานเสือภูเขา	คอร์ส bikini wax	0.125

```
In [98]: links= links.loc[(links['item'] !=links['related item'])].sort_values('value',ascending
links.shape
```

```
Out[98]: (3540, 3)
```

```
In [99]: links_filtered=links.loc[
(links['item'] != links['related item'])& (links['value'] > 0.
links_filtered.shape
```

Out[99]: (230, 3)

```
In [100... links_filtered.sort_values(by = ['value'], ascending = False).head(10)
```

```
Out[100...
      item  related item  value
1892   กระเป๋          ชานู  1.000
1919   กระเป๋   ครีมกันแดด  1.000
1979   ชานู      ครีมกันแดด  1.000
3571   ครีมกันแดด   กระเป๋  1.000
1951   ชานู      กระเป๋  1.000
3572   ครีมกันแดด   ชานู  1.000
2759   โรดี      ครีมกันแดด  0.978
2732   โรดี          ชานู  0.978
1965   ชานู      โรดี  0.978
1905   กระเป๋      โรดี  0.978
```

Graph network show the relationship on each pair of items by similarity score

We could recommend items based on item similarity. Filtering the similarity using 0.75 threshold, and visualize them in graph. A node represents an item. Edge thickness and color represent cosine similarity between 2 items.

```
In [101... fig, ax =plt.subplots(figsize=(25,18))
GA=nx.from_pandas_edgelist(links_filtered,source='item',target='related item',edge_attr
#GA.nodes()
# nx.draw(GA,with_labels=True,node_color='orange')
# plt.show()
weight2 = [ float(i['value']) for i in dict(GA.edges).values()]
weight2 = ((np.array(weight2)- min(weight2))/(max(weight2)-min(weight2)))*5
labels2 = [i for i in dict(GA.nodes)]
labels2 = {i:i for i in dict(GA.nodes).keys()}
#pos = nx.spring_layout(GA)
pos = nx.spring_layout(GA,weight='weight2', k=12)
nx.draw_networkx_nodes(GA, pos,ax = ax,node_color = '#ca002c') # #e636e1 ##f9c3ff
nx.draw_networkx_edges(GA, pos,edge_color='#7692f6',arrowsize=10, width=weight2, ax=ax
edge_labels = nx.get_edge_attributes(GA, 'value')
#edge_labels = nx.get_edge_attributes(G, 'lift')
nx.draw_networkx_edge_labels(GA, pos, edge_labels=edge_labels, font_color='red')#, **l
_ = nx.draw_networkx_labels(GA, pos, labels2, ax=ax)
plt.savefig('collaborative.png',format = 'png',dpi = 150)
```

