fp-tree

October 3, 2023

Name: Ankush Kumar Reg: AP21110011026 CSE 'P'

[1]:
```
import pandas as pd
from itertools import product
from itertools import combinations
df=pd.DataFrame({
    "TId":['T100','T200','T300','T400','T500','T600','T700','T800','T900'],
    "Item_Ids":
  ↪[['I1','I2','I5'],['I2','I4'],['I2','I3'],['I1','I2','I4'],['I1','I3'],['I2','I3'],['I1','I
})
msp=2
df
```

[1]:
```
   TId          Item_Ids
0  T100      [I1, I2, I5]
1  T200          [I2, I4]
2  T300          [I2, I3]
3  T400      [I1, I2, I4]
4  T500          [I1, I3]
5  T600          [I2, I3]
6  T700          [I1, I3]
7  T800  [I1, I2, I3, I5]
8  T900      [I1, I2, I3]
```

[5]:
```
flattened_list = [item for sublist in df['Item_Ids'].values.tolist() for item␣
  ↪in sublist]
ids = sorted(list(set(flattened_list)))
itemset=pd.DataFrame(columns=['Itemset','Count'])
for i in range(len(ids)):
    itemset.loc[i] = [ids[i],df['Item_Ids'].apply(lambda x: ids[i] in x).sum()]
print('Count of each Item: ')
print(itemset)
print('Sorted and Filtered ')
itemset=itemset[itemset['Count']>=msp].sort_values(by='Count',ascending=False)
print(itemset)
```

```
Count of each Item:
  Itemset  Count
```

```
0      I1      6
1      I2      7
2      I3      6
3      I4      2
4      I5      2
Sorted and Filtered
  Itemset  Count
1      I2      7
0      I1      6
2      I3      6
3      I4      2
4      I5      2
```

[6]:

```
df['Item_Ids'] = df['Item_Ids'].apply(lambda x: sorted(x, key=lambda item:␣
 ↪itemset[itemset['Itemset'] == item]['Count'].values[0], reverse=True))
print(df)
```

```
   TId          Item_Ids
0  T100      [I2, I1, I5]
1  T200          [I2, I4]
2  T300          [I2, I3]
3  T400      [I2, I1, I4]
4  T500          [I1, I3]
5  T600          [I2, I3]
6  T700          [I1, I3]
7  T800  [I2, I1, I3, I5]
8  T900      [I2, I1, I3]
```

[6]:

```
import pandas as pd
import pyfpgrowth

df = pd.DataFrame({
    "TId":['T100','T200','T300','T400','T500','T600','T700','T800','T900'],
    "Item_Ids":
 ↪[['I1','I2','I5'],['I2','I4'],['I2','I3'],['I1','I2','I4'],['I1','I3'],['I2','I3'],['I1','I
})

transactions = df['Item_Ids'].tolist()

min_support = 2

patterns = pyfpgrowth.find_frequent_patterns(transactions, min_support)

for itemset, support in patterns.items():
    print(f"Itemset: {itemset}, Support: {support}")
```

```
Itemset: ('I5',), Support: 2
Itemset: ('I1', 'I5'), Support: 2
```

```
Itemset: ('I2', 'I5'), Support: 2
Itemset: ('I1', 'I2', 'I5'), Support: 2
Itemset: ('I4',), Support: 2
Itemset: ('I2', 'I4'), Support: 2
Itemset: ('I1',), Support: 6
Itemset: ('I1', 'I2'), Support: 4
Itemset: ('I2', 'I3'), Support: 4
Itemset: ('I1', 'I2', 'I3'), Support: 2
Itemset: ('I1', 'I3'), Support: 4
Itemset: ('I2',), Support: 7
```

[4]:

```
patterns
```

[4]:

```
{('I5',): 2,
 ('I1', 'I5'): 2,
 ('I2', 'I5'): 2,
 ('I1', 'I2', 'I5'): 2,
 ('I4',): 2,
 ('I2', 'I4'): 2,
 ('I1',): 6,
 ('I1', 'I2'): 4,
 ('I2', 'I3'): 4,
 ('I1', 'I2', 'I3'): 2,
 ('I1', 'I3'): 4,
 ('I2',): 7}
```

[7]:

```
type(patterns)
```

[7]:

```
dict
```

[ ]:

```

```