```
try:
    from google.colab import drive
    %tensorflow_version 2.x
    COLAB = True
    print("Assignment 7")
    print("Note: using Google CoLab")
except:
    print("Assignment 7")
    print("Note: not using Google CoLab")
    COLAB = False

print("Rohit Byas", "\nRoll Number: 181210043")
```

```
Assignment 7
Note: using Google CoLab
Rohit Byas
Roll Number: 181210043
```

```
!pip install efficient-apriori
!pip install pyfpgrowth
```

```
Requirement already satisfied: efficient-apriori in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: pyfpgrowth in /usr/local/lib/python3.6/dist-packages (1.0
```

```
from efficient_apriori import apriori
import pyfpgrowth
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
```

Question 1:

The Apriori Algorithm was Proposed by Agrawal et al in 1993. It is an important data mining model studied extensively by the database and data mining community. Assume all data are categorical. Consider the following transactions, records from a supermarket, First solve this by hand :

Transaction ID - Item Purchased

- 1 - Cheese, Juice, Milk
- 2 - Bread, Juice, Milk
- 3 - Bread, Cheese, Egg, Juice
- 4 - Bread, Milk, Yogurt
- 5 - Bread, Cheese, Juice

1. Write a python program to implement Apriori Algorithm for the above transactions. Consider the support=50 % and confidence=60 %.

   (a) using python module

   (b) without using module

2. Write a python program to implement FP Growth algorithm.

   (a) using python module

   (b) without using module

```
#Applying Apriori Algo
dataset = [['Milk', 'Cheese', 'Juice'],
           ['Bread', 'Juice', 'Milk'],
           ['Bread', 'Cheese', 'Egg', 'Juice'],
           ['Bread', 'Milk', 'Yogurt'],
           ['Bread', 'Cheese', 'Juice']]

itemsets1, rules1 = apriori(dataset, min_support=0.5, min_confidence=0.6)
for j in rules1:
  print(j)
```

```
    {Juice} -> {Cheese} (conf: 0.750, supp: 0.600, lift: 1.250, conv: 1.600)
    {Cheese} -> {Juice} (conf: 1.000, supp: 0.600, lift: 1.250, conv: 200000000.000)
    {Juice} -> {Bread} (conf: 0.750, supp: 0.600, lift: 0.938, conv: 0.800)
    {Bread} -> {Juice} (conf: 0.750, supp: 0.600, lift: 0.938, conv: 0.800)
```

```
#Applying frquent pattern algo
patterns = pyfpgrowth.find_frequent_patterns(dataset, 2)
fp1 = pyfpgrowth.generate_association_rules(patterns, 0.6)

print(fp1)
```

```
    {('Cheese',): (('Bread', 'Juice'), 0.6666666666666666), ('Juice',): (('Bread',), 0.75),
```

Question 2:

Compare Apriori and FP Growth Algorithm on the following input. Mention and discuss your observation.

- transactions =

```
[['MILK', 'BREAD', "BISCUIT"],
 ['BREAD', 'MILK', 'BISCUIT', "CORNFLAKES"],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['JAM', 'MAGGI', 'BREAD', 'MILK'],
```

```
                    ['MAGGI', 'TEA', 'BISCUIT'],
                    ['BREAD', 'TEA', 'BOURNVITA'],
                    ['MAGGI', 'TEA', 'CORNFLAKES'],
                    ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
                    ['JAM', 'MAGGI', 'BREAD', 'TEA'],
                    ['BREAD', 'MILK'],
                    ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
                    ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
                    ['COFFEE', 'SUGAR', 'BOURNVITA'],
                    ['BREAD', 'COFFEE', 'COCK'],
                    ['BREAD', 'SUGER', 'BISCUIT'],
                    ['COFFEE', 'SUGER', 'CORNFLAKES'],
                    ['BREAD', 'SUGER', 'BOURNVITA'],
                    ['BREAD', 'COFFEE', 'SUGER'],
                    ['BREAD', 'COFFEE', 'SUGER'],
                    ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]


transactions = [['MILK', 'BREAD', "BISCUIT"],
                ['BREAD', 'MILK', 'BISCUIT', "CORNFLAKES"],
                ['BREAD', 'TEA', 'BOURNVITA'],
                ['JAM', 'MAGGI', 'BREAD', 'MILK'],
                ['MAGGI', 'TEA', 'BISCUIT'],
                ['BREAD', 'TEA', 'BOURNVITA'],
                ['MAGGI', 'TEA', 'CORNFLAKES'],
                ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
                ['JAM', 'MAGGI', 'BREAD', 'TEA'],
                ['BREAD', 'MILK'],
                ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
                ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
                ['COFFEE', 'SUGAR', 'BOURNVITA'],
                ['BREAD', 'COFFEE', 'COCK'],
                ['BREAD', 'SUGER', 'BISCUIT'],
                ['COFFEE', 'SUGER', 'CORNFLAKES'],
                ['BREAD', 'SUGER', 'BOURNVITA'],
                ['BREAD', 'COFFEE', 'SUGER'],
                ['BREAD', 'COFFEE', 'SUGER'],
                ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]

itemsets2, rules2 = apriori(transactions, min_support=0.2, min_confidence=0.6)

for i in rules2:
  print(i)

    {MILK} -> {BREAD} (conf: 0.800, supp: 0.200, lift: 1.231, conv: 1.750)
    {MAGGI} -> {TEA} (conf: 0.800, supp: 0.200, lift: 2.286, conv: 3.250)
```

```
{CORNFLAKES} -> {COFFEE} (conf: 0.667, supp: 0.200, lift: 1.667, conv: 1.800)
{SUGER} -> {BREAD} (conf: 0.800, supp: 0.200, lift: 1.231, conv: 1.750)
```

```
patterns1 = pyfpgrowth.find_frequent_patterns(transactions, 2)
fp2 = pyfpgrowth.generate_association_rules(patterns1, 0.5)
print(fp2)
```

```
{('JAM',): (('BREAD', 'MAGGI'), 1.0), ('BREAD', 'JAM'): (('MAGGI',), 1.0), ('BREAD', 'MA
```