Demonstration and Comparative Analysis of Apriori and FP Growth Algorithm
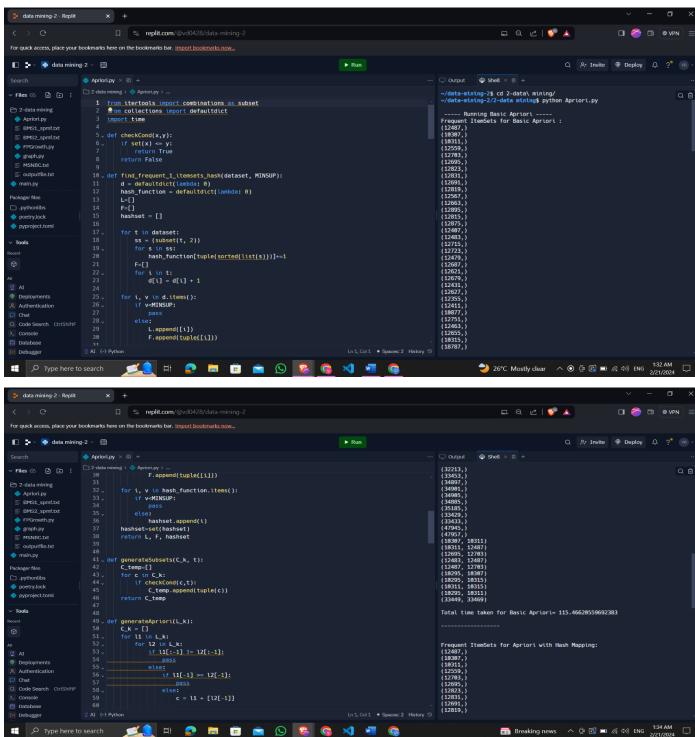
DOCUMENT FOR REFERENCE##

Files:
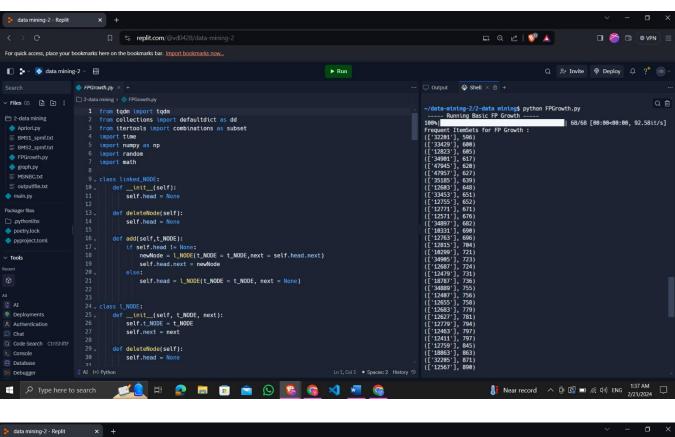
1. Apriori.py: python file containing code for Apriori algorithm.
2. FPGrowth.py: python file containing code for Frequent Pattern Growth algorithm.
3. graph.py: python file containing code for generating comparison graph between   both the algorithms.
4. BMS1_spmf.txt: Input file one containing data set 1 in text form.
5. BMS2_spmf.txt: Input file two containing data set 2 in text form.
6. MSNBC.txt: Input file three containing data set in 3 text form.
7. output for Apriori.txt: Text file containing output of Apriori.py file.
8. output for FPGrowth.txt: Text file containing output of FPGrowth.py file.
9. Screenshot (223).png: Image containing output of graph.py file.

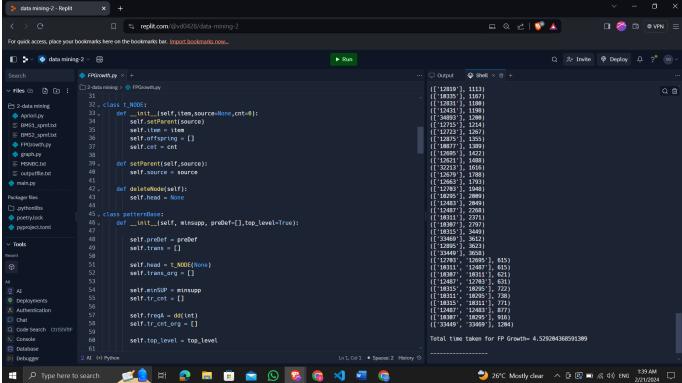10. outputfile.txt: Text file containing comparison data between both algorithms for different minimum support values.
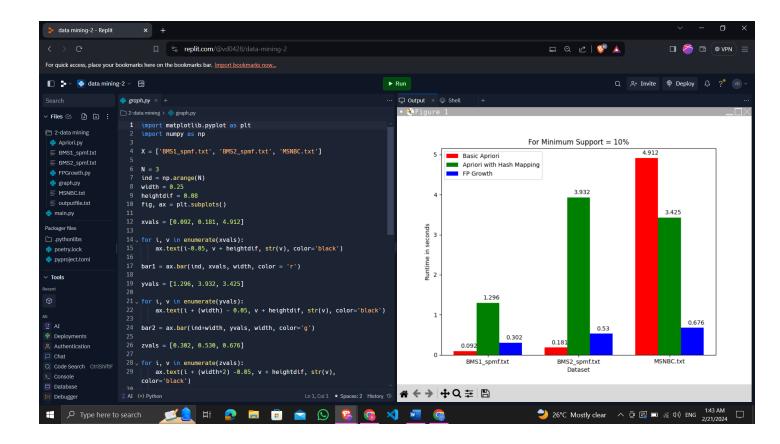
Screenshot of codes and output:

Apriori.py:

```python
from itertools import combinations as subset
from collections import defaultdict
import time

def checkCond(x,y):
    if set(x) <= y:
        return True
    return False

def find_frequent_1_itemsets_hash(dataset, MINSUP):
    d = defaultdict(lambda: 0)
    hash_function = defaultdict(lambda: 0)
    L=[]
    F=[]
    hashset = []

    for t in dataset:
        ss = (subset(t, 2))
        for s in ss:
            hash_function[tuple(sorted(list(s)))]+=1
        F=[]
        for i in t:
            d[i] = d[i] + 1

        for i, v in d.items():
            if v<MINSUP:
                pass
            else:
                L.append([i])
                F.append(tuple([i]))
```

```
~/data-mining-2$ cd 2-data\ mining/
~/data-mining-2/2-data mining$ python Apriori.py

----- Running Basic Apriori -----
Frequent ItemSets for Basic Apriori :
(12487,)
(10307,)
(10311,)
(12559,)
(12703,)
(12695,)
(12823,)
(12831,)
(12691,)
(12819,)
(12567,)
(12663,)
(12895,)
(12815,)
(12875,)
(12407,)
(12483,)
(12715,)
(12723,)
(12479,)
(12687,)
(12621,)
(12679,)
(12431,)
(12627,)
(12355,)
(12411,)
(10877,)
(12751,)
(12463,)
(12655,)
(10315,)
(18787,)
```

```python
                F.append(tuple([i]))

        for i, v in hash_function.items():
            if v<MINSUP:
                pass
            else:
                hashset.append(i)
    hashset=set(hashset)
    return L, F, hashset

def generateSubsets(C_k, t):
    C_temp=[]
    for c in C_k:
        if checkCond(c,t):
            C_temp.append(tuple(c))
    return C_temp


def generateApriori(L_k):
    C_k = []
    for l1 in L_k:
        for l2 in L_k:
            if l1[:-1] != l2[:-1]:
                pass
            else:
                if l1[-1] >= l2[-1]:
                    pass
                else:
                    c = l1 + [l2[-1]]
```

```
(32213,)
(33453,)
(34897,)
(34901,)
(34905,)
(34885,)
(35185,)
(33429,)
(33433,)
(47945,)
(47957,)
(10307, 10311)
(10311, 12487)
(12695, 12703)
(12483, 12487)
(12487, 12703)
(10295, 10307)
(10295, 10315)
(10311, 10315)
(10295, 10311)
(33449, 33469)

Total time taken for Basic Apriori= 115.46620559692383

-------------------

Frequent ItemSets for Apriori with Hash Mapping:
(12487,)
(10307,)
(10311,)
(12559,)
(12703,)
(12823,)
(12831,)
(12691,)
(12819,)
```

Frequent Pattern Growth Algorithm:

Graph.py:

Final result obtained:

For Minimum Support = 1%:

BMS1_spmf.txt:

Total time taken for Basic Apriori= 74.7482898235321 seconds

Total time taken for Apriori with Hash Mapping= 1.3318769931793213 seconds

Total time taken for FP Growth= 1.0110151767730713 seconds

BMS2_spmf.txt:

Total time taken for Basic Apriori= 66.2326180934906 seconds

Total time taken for Apriori with Hash Mapping= 5.244940757751465 seconds

Total time taken for FP Growth= 1.2578978538513184 seconds

MSNBC.txt:

Total time taken for Basic Apriori= 153.68514347076416 seconds

Total time taken for Apriori with Hash Mapping= 154.7875907421112 seconds

Total time taken for FP Growth= 1.497495412826538 seconds

For Minimum Support = 5%:

BMS1_spmf.txt:

Total time taken for Basic Apriori= 0.28157925605773926 seconds

Total time taken for Apriori with Hash Mapping= 1.2909183502197266 seconds

Total time taken for FP Growth= 0.3462948799133301 seconds

BMS2_spmf.txt

Total time taken for Basic Apriori= 0.18675875663757324 seconds

Total time taken for Apriori with Hash Mapping= 3.8645527362823486 seconds

Total time taken for FP Growth= 0.5378656387329102 seconds

MSNBC.txt

Total time taken for Basic Apriori= 12.381566762924194 seconds

Total time taken for Apriori with Hash Mapping= 10.221471309661865 seconds

Total time taken for FP Growth= 0.8272049427032471 seconds

For Minimum Support = 10%:

BMS1_spmf.txt

Total time taken for Basic Apriori= 0.09195780754089355 seconds

Total time taken for Apriori with Hash Mapping= 1.2956795692443848 seconds

Total time taken for FP Growth= 0.30190372467041016 seconds

BMS2_spmf.txt

Total time taken for Basic Apriori= 0.1815943717956543 seconds

Total time taken for Apriori with Hash Mapping= 3.9320201873779297 seconds

Total time taken for FP Growth= 0.5301475524902344 seconds

MSNBC.txt

Total time taken for Basic Apriori= 4.912920951843262 seconds

Total time taken for Apriori with Hash Mapping= 3.4259331226348877 seconds

Total time taken for FP Growth= 0.676424503326416 seconds

~VIKRANT KUMAR VERMA (RA2111003010522)