Advanced Tutorial on Implementing FFIMiner Algorithm

In this tutorial, we explain how the FFIMiner algorithm can be implemented by varying the minimum support values

Step 1: Import the FFIMiner algorithm and pandas data frame

In [1]:
```
from PAMI.fuzzyFrequentPatterns.basic import FFIMiner  as alg
import pandas as pd
```

Step 2: Specify the following input parameters

In [2]:
```
inputFile = 'T10_utility.txt'

minimumSupportCountList=[800,1000,1200,1400,1800]  #Users can also specify this cons
seperator=' '
result = pd.DataFrame(columns=['algorithm', 'minSup', 'patterns', 'runtime', 'memory
#initialize a data frame to store the results of FFIMiner algorithm
```

Step 3: Execute the FFIMiner algorithm using a for loop

In [3]:
```
algorithm = 'FFIMiner'  #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.FFIMiner(iFile=inputFile, minSup=minSupCount, sep=seperator)
    obj.startMine()
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj.getPatterns()), c
```

In [4]:
```
print(result)
```

```
  algorithm  minSup  patterns     runtime     memory
0  FFIMiner     800       468  414.086585  435793920
1  FFIMiner    1000       383  316.264430  427044864
2  FFIMiner    1200       318  243.798982  416133120
3  FFIMiner    1400       259  180.152071  403210240
4  FFIMiner    1800       174   98.035970  380014592
```

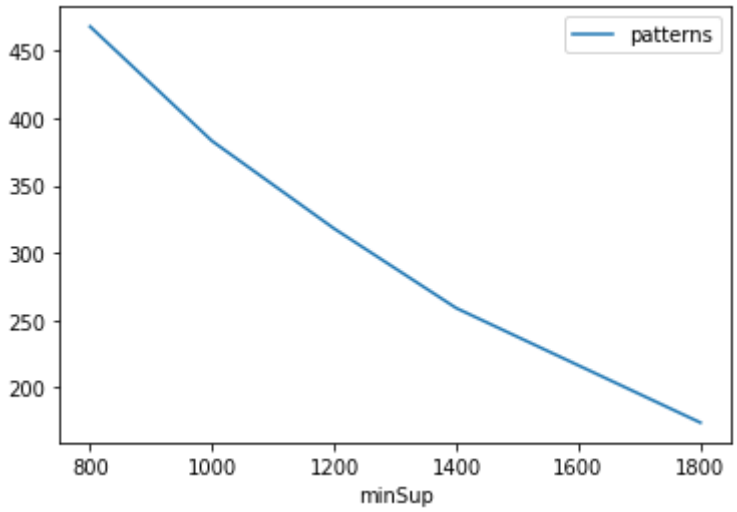
Step 5: Visualizing the results

Step 5.1 Importing the plot library

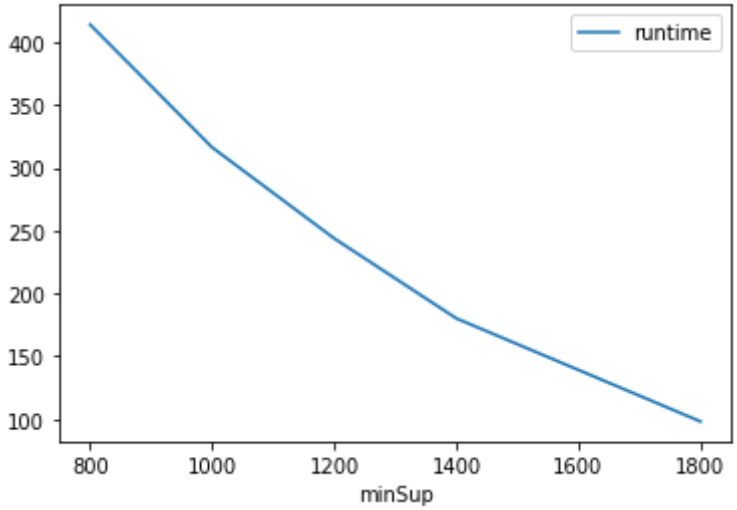
In [5]:
```
from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
```

Step 5.2. Plotting the number of patterns

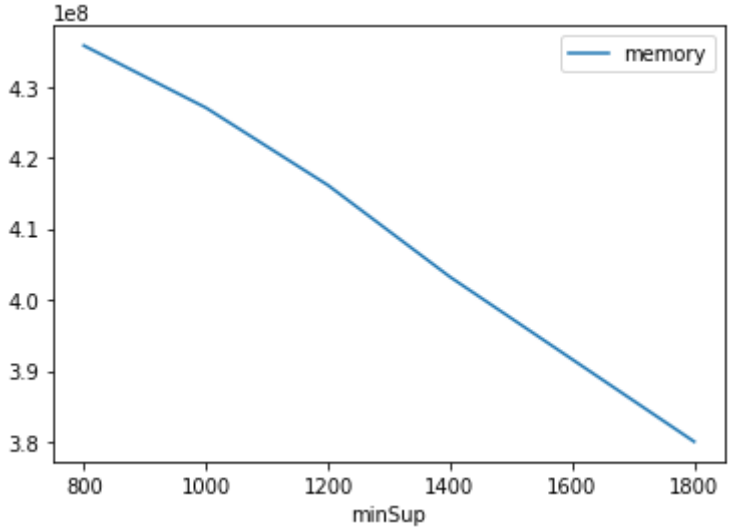
In [6]:
```
ab = plt.plotGraphsFromDataFrame(result)
ab.plotGraphsFromDataFrame() #drawPlots()
```



Graph for No Of Patterns is successfully generated!



Graph for Runtime taken is successfully generated!



Graph for memory consumption is successfully generated!

Step 6: Saving the results as latex files

In [7]:
```
from PAMI.extras.graph import generateLatexFileFromDataFrame as gdf
gdf.generateLatexCode(result)
```

Latex files generated successfully