Advanced Tutorial on Implementing PFS_ECLAT Algorithm

In this tutorial, we explain how the Periodic Frequent Spatial ECLAT (PFS_ECLAT) algorithm can be implemented by varying the minimum support values

Step 1: Import the PFS_ECLAT algorithm and pandas data frame

In [1]:

```
from PAMI.periodicFrequentSpatialPattern import PFS_ECLAT  as alg
import pandas as pd
```

Step 2: Specify the following input parameters

In [2]:

```
inputFile = 'temporal_T10I4D100K.csv'
seperator='¥t'
maxmunPeriodCount=5000
minimumSupportCountList = [100, 150, 200, 250, 300]
#minimumSupport can also specified between 0 to 1. E.g., minSupList = [0.005, 0.006,
neighborFile='T10_utility_neighbour.txt'
result = pd.DataFrame(columns=['algorithm', 'minSup', 'maxPer','patterns', 'runtime'
#initialize a data frame to store the results of PFS_ECLAT algorithm
```

Step 3: Execute the PFS_ECLAT algorithm using a for loop

In [3]:

```
algorithm = 'PFS_ECLAT'  #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.PFS_ECLAT(iFile=inputFile, minSup=minSupCount,maxPer=maxmunPeriodCoun
    obj.startMine()
    neighborFile='T10_utility_neighbour.txt'
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount,maxmunPeriodCount, len(obj
```

```
100 5000
Spatial Periodic Frequent patterns were generated successfully using SpatialEclat al
gorithm
150 5000
Spatial Periodic Frequent patterns were generated successfully using SpatialEclat al
gorithm
200 5000
Spatial Periodic Frequent patterns were generated successfully using SpatialEclat al
gorithm
250 5000
Spatial Periodic Frequent patterns were generated successfully using SpatialEclat al
gorithm
300 5000
Spatial Periodic Frequent patterns were generated successfully using SpatialEclat al
gorithm
```

In [4]:

```
print(result)
```

```
    algorithm  minSup  maxPer  patterns    runtime     memory
0  PFS_ECLAT     100    5000      4997  15.676548  247517184
1  PFS_ECLAT     150    5000      3733  12.620347  247271424
2  PFS_ECLAT     200    5000      2918  11.538911  246800384
3  PFS_ECLAT     250    5000      2123  10.567986  246669312
4  PFS_ECLAT     300    5000      1642   9.954227  246595584
```

Step 5: Visualizing the results

Step 5.1 Importing the plot library

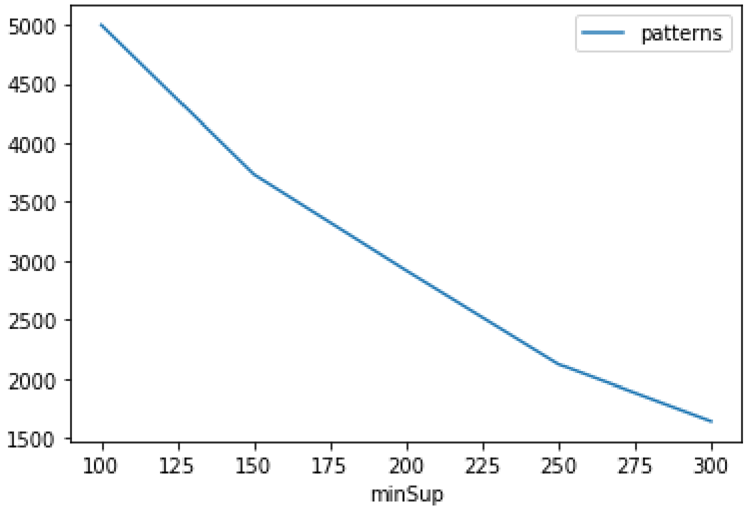
In [5]:

```
from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
```

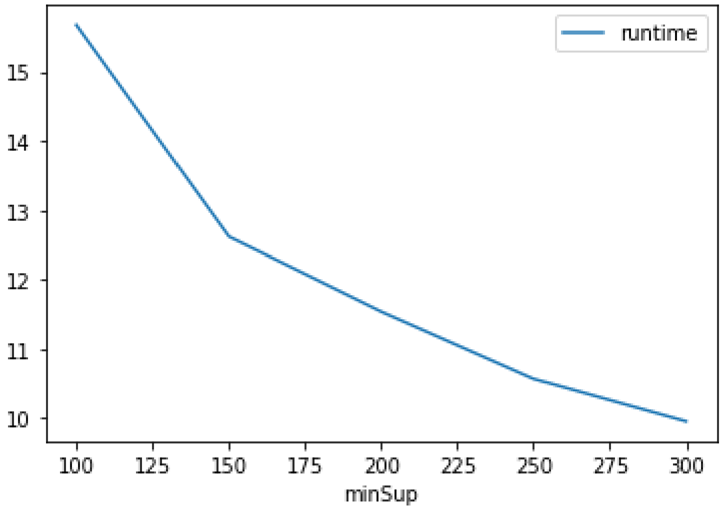
Step 5.2. Plotting the number of patterns

In [6]:

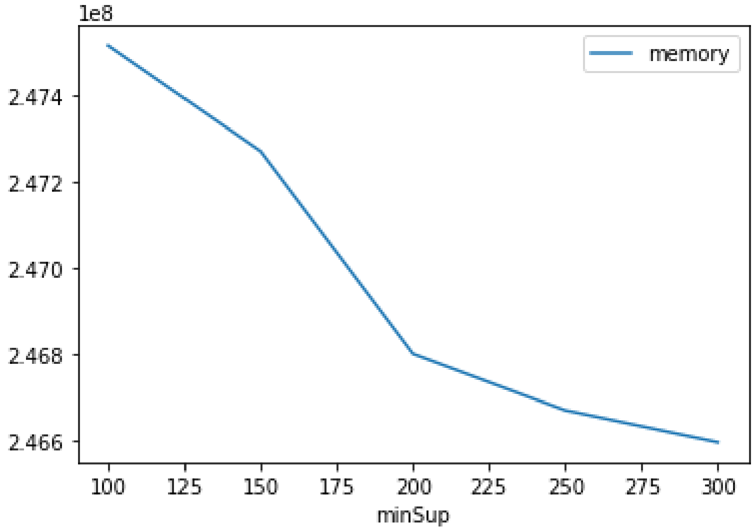
```
ab = plt.plotGraphsFromDataFrame(result)
ab.plotGraphsFromDataFrame() #drawPlots()
```



```
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
```