Advanced Tutorial on Implementing SpatialECLAT Algorithm

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

Step 1: Import the SpatialECLAT algorithm and pandas data frame

In [1]:
```
from PAMI.frequentSpatialPattern.basic import SpatialECLAT  as alg
import pandas as pd
```

Step 2: Specify the following input parameters

In [2]:
```
inputFile = 'transactional_T10I4D100K.csv'
seperator='¥t'
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', 'patterns', 'runtime', 'memory
#initialize a data frame to store the results of SpatialECLAT algorithm
```

Step 3: Execute the SpatialECLAT algorithm using a for loop

In [3]:
```
algorithm = 'SpatialECLAT'  #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.SpatialECLAT('transactional_T10I4D100K.csv', minSup=minSupCount,nFile=
    obj.startMine()
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj.getPatterns()), o
```

```
Spatial Frequent patterns were generated successfully using SpatialECLAT algorithm
Spatial Frequent patterns were generated successfully using SpatialECLAT algorithm
Spatial Frequent patterns were generated successfully using SpatialECLAT algorithm
Spatial Frequent patterns were generated successfully using SpatialECLAT algorithm
Spatial Frequent patterns were generated successfully using SpatialECLAT algorithm
```

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

```
      algorithm  minSup  patterns    runtime     memory
0  SpatialECLAT     100      4300  39.124038  244211712
1  SpatialECLAT     150      2971  23.278850  244662272
2  SpatialECLAT     200      2291  18.359012  244826112
3  SpatialECLAT     250      1798  15.450596  244842496
4  SpatialECLAT     300      1440  13.883977  244600832
```

Step 5: Visualizing the results

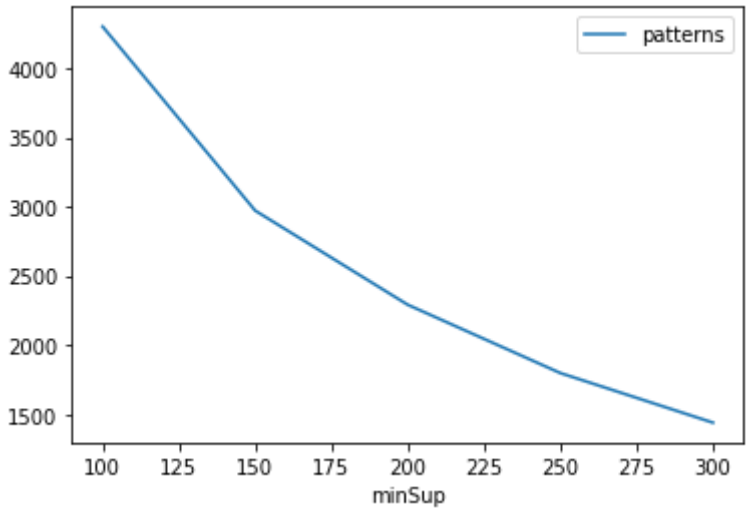
Step 5.1 Importing the plot library

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

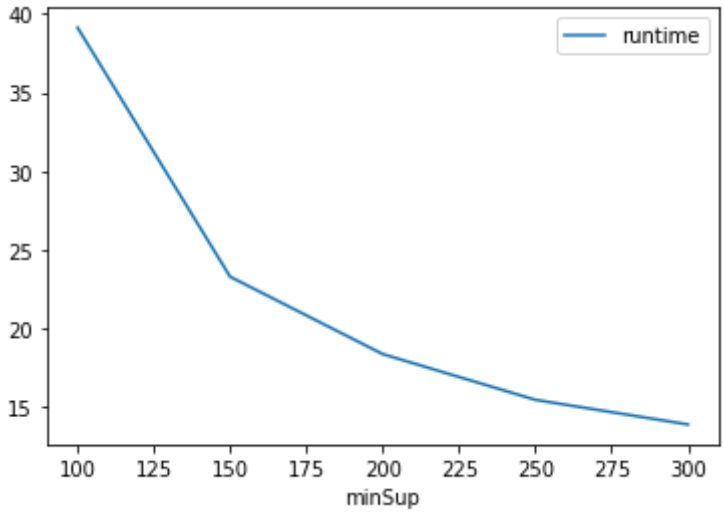
Step 5.2. Plotting the number of patterns

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

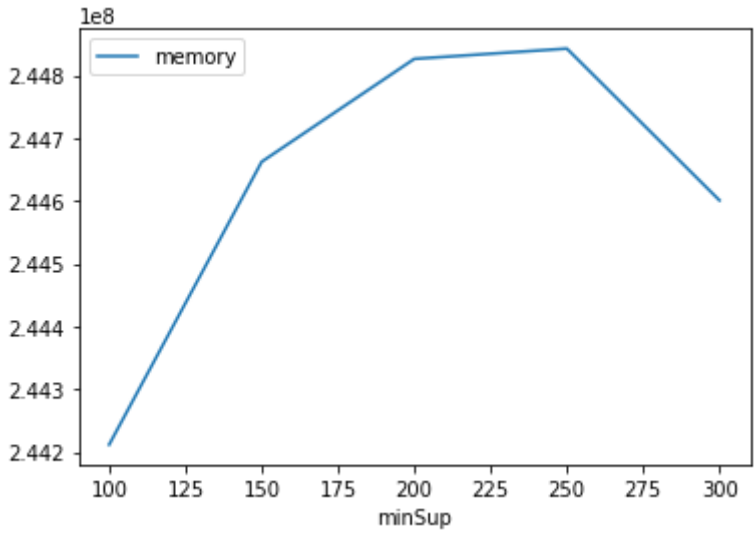
```
ab.plotGraphsFromDataFrame() #drawPlots()
```



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