Advanced Tutorial on Implementing CPGrowthPlus Algorithm

In this tutorial, we explain how the Correlated Pattern GrowthPlus (CPGrowthPlus) algorithm can be implemented by varying the minimum support values

Step 1: Import the CPGrowthPlus algorithm and pandas data frame

In [1]:
```
from PAMI.correlatedPattern.basic import CPGrowthPlus  as alg
import pandas as pd
```

Step 2: Specify the following input parameters

In [2]:
```
inputFile = 'transactional_T10I4D100K.csv'
seperator='¥t'
minAllConfCount=0.1
minimumSupportCountList = [100, 150, 200, 250, 300]
#minimumSupport can also specified between 0 to 1. E.g., minSupList = [0.005, 0.006,

result = pd.DataFrame(columns=['algorithm', 'minSup',"minAllConf" , 'patterns', 'rur
#initialize a data frame to store the results of CPGrowthPlus algorithm
```

Step 3: Execute the CPGrowthPlus algorithm using a for loop

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

```
Correlated Frequent patterns were generated successfully using CorrelatedPatternGrow
th algorithm
Correlated Frequent patterns were generated successfully using CorrelatedPatternGrow
th algorithm
Correlated Frequent patterns were generated successfully using CorrelatedPatternGrow
th algorithm
Correlated Frequent patterns were generated successfully using CorrelatedPatternGrow
th algorithm
Correlated Frequent patterns were generated successfully using CorrelatedPatternGrow
th algorithm
```

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

```
      algorithm  minSup  minAllConf  patterns    runtime     memory
0  CPGrowthPlus     100         0.1      5758  13.352987  402366464
1  CPGrowthPlus     150         0.1     15301  12.484414  466128896
2  CPGrowthPlus     200         0.1     22335  13.368088  484454400
3  CPGrowthPlus     250         0.1     28536  13.717880  490815488
4  CPGrowthPlus     300         0.1     33074  14.665221  499810304
```

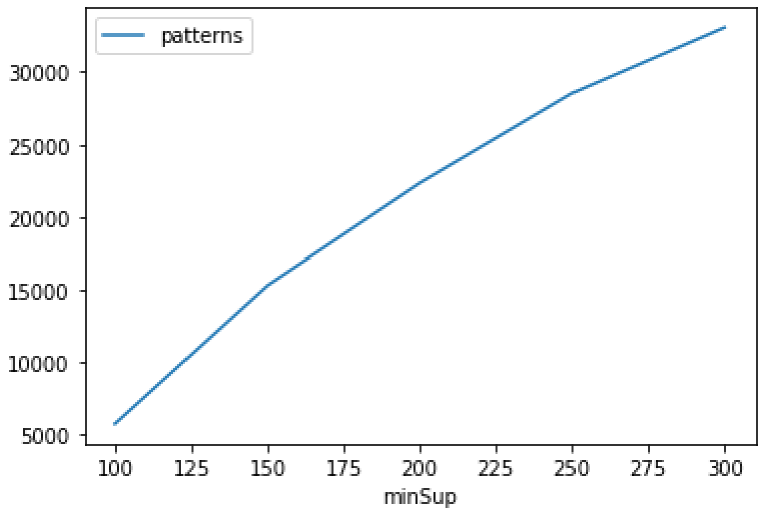
Step 5: Visualizing the results

Step 5.1 Importing the plot library

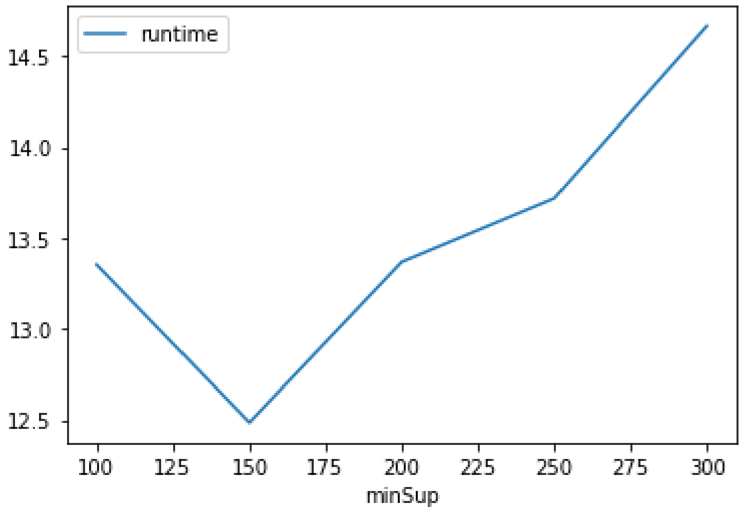
In [5]:
```
from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
```

Step 5.2. Plotting the number of patterns

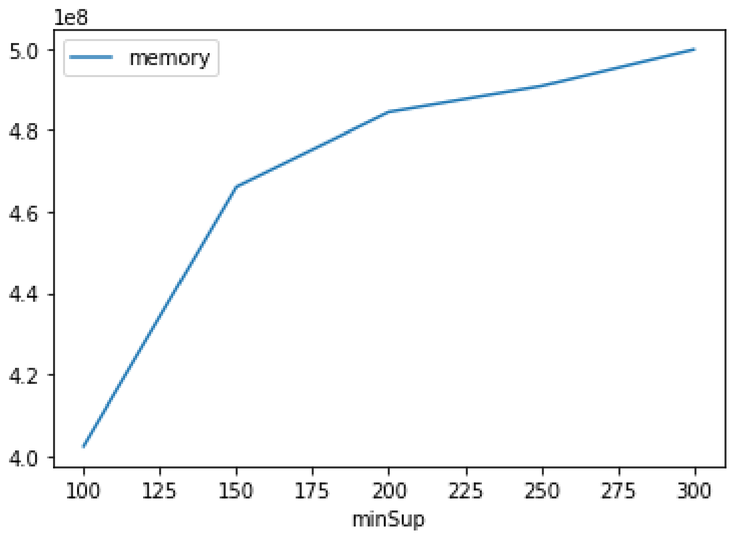
In [6]:
```
ab = plt.plotGraphsFromDataFrame(result)
ab.plotGraphsFromDataFrame() #drawPlots()
```



```
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
```

In [ ]:
```
```