Advanced Tutorial on Implementing ECLATDiffset Algorithm

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

Step 1: Import the ECLATDiffset algorithm and pandas data frame

In [1]:
```
from PAMI.frequentPattern.basic import ECLATDiffset  as alg
import pandas as pd
```

Step 2: Specify the following input parameters

In [2]:
```
inputFile = 'transactional_T10I4D100K.csv'
seperator='¥t'
minimumSupportCountList = [1000, 1500, 2000, 2500, 3000]
#minimumSupport can also specified between 0 to 1. E.g., minSupList = [0.005, 0.006,

result = pd.DataFrame(columns=['algorithm', 'minSup', 'patterns', 'runtime', 'memory
#initialize a data frame to store the results of ECLATDiffset algorithm
```

Step 3: Execute the ECLATDiffset algorithm using a for loop

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

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

Step 5: Visualizing the results

Step 5.1 Importing the plot library

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

Step 5.2. Plotting the number of patterns

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

Step 6: Saving the results as latex files

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

In [ ]:
```

```